Université de Montréal

Urban Detection From Hyperspectral Images Using Dimension-Reduction Model and Fusion of Multiple Segmentations Based on Stuctural and Textural Features

par
Jin He

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la Faculté des arts et des sciences
en vue de l'obtention du grade de Maître ès sciences (M.Sc.)
en informatique

Septembre, 2013

Université de Montréal
Faculté des arts et des sciences

Ce mémoire intitulé:

Urban Detection From Hyperspectral Images Using Dimension-Reduction Model and Fusion of Multiple Segmentations Based on Stuctural and Textural Features

présenté par:

Jin He

a été évalué par un jury composé des personnes suivantes:

Jean Meunier
président-rapporteur

Max Mignotte
directeur de recherche

Neil Stewart
membre du jury

Mémoire accepté le 10 septembre 2013

RÉSUMÉ

Ce mémoire de maîtrise présente une nouvelle approche non supervisée pour détecter et segmenter les régions urbaines dans les images hyperspectrales. La méthode proposée nécessite trois étapes. Tout d'abord, afin de réduire le coût calculatoire de notre algorithme, une image couleur du contenu spectral est estimée. A cette fin, une étape de réduction de dimensionalité non-linéaire, basée sur deux critères complémentaires mais contradictoires de bonne visualisation; à savoir la précision et le contraste, est réalisée pour l'affichage couleur de chaque image hyperspectrale. Ensuite, pour discriminer les régions urbaines des régions non urbaines, la seconde étape consiste à extraire quelques caractéristiques discriminantes (et complémentaires) sur cette image hyperspectrale couleur. A cette fin, nous avons extrait une série de paramètres discriminants pour décrire les caractéristiques d'une zone urbaine, principalement composée d'objets manufacturés de formes simples géométriques et régulières. Nous avons utilisé des caractéristiques texturales basées sur les niveaux de gris, la magnitude du gradient ou des paramètres issus de la matrice de co-occurrence combinés avec des caractéristiques structurelles basées sur l'orientation locale du gradient de l'image et la détection locale de segments de droites. Afin de réduire encore la complexité de calcul de notre approche et éviter le problème de la "malédiction de la dimensionnalité" quand on décide de regrouper des données de dimensions élevées, nous avons décidé de classifier individuellement, dans la derniére étape, chaque caractéristique texturale ou structurelle avec une simple procédure de K-moyennes et ensuite de combiner ces segmentations grossières, obtenues à faible coût, avec un modèle efficace de fusion de cartes de segmentations. Les expérimentations données dans ce rapport montrent que cette stratégie est efficace visuellement et se compare favorablement aux autres méthodes de détection et segmentation de zones urbaines à partir d'images hyperspectrales.

Mots clés:

images AVIRIS, détection et segmentation de régions urbaines, fusion de seg-

mentations, gradient, matrice de co-occurrence, image hyperspectrale, K-moyennes, détection de segment de droite, réduction de dimensionnalité non-linéaire.

ABSTRACT

This master's thesis presents a new approach to urban area detection and segmentation in hyperspectral images. The proposed method relies on a three-step procedure. First, in order to decrease the computational complexity, an informative three-color composite image, minimizing as much as possible the loss of information of the spectral content, is computed. To this end, a non-linear dimensionality reduction step, based on two complementary but contradictory criteria of good visualization, namely accuracy and contrast, is achieved for the color display of each hyperspectral image. In order to discriminate between urban and non-urban areas, the second step consists of extracting some complementary and discriminant features on the resulting (three-band) color hyperspectral image. To attain this goal, we have extracted a set of features relevant to the description of different aspects of urban areas, which are mainly composed of man-made objects with regular or simple geometrical shapes. We have used simple textural features based on grey-levels, gradient magnitude or grey-level co-occurence matrix statistical parameters combined with structural features based on gradient orientation, and straight segment detection. In order to also reduce the computational complexity and to avoid the so-called "curse of dimensionality" when clustering high-dimensional data, we decided, in the final third step, to classify each individual feature (by a simple K-means clustering procedure) and to combine these multiple low-cost and rough image segmentation results with an efficient fusion model of segmentation maps. The experiments reported in this report demonstrate that the proposed segmentation method is efficient in terms of visual evaluation and performs well compared to existing and automatic detection and segmentation methods of urban areas from hyperspectral images.

Keywords:

AVIRIS images, detection, features extraction, fusion of label field, gradient, GLCM, hyperspectral image, K-means clustering, line segment detector, non-linear

dimension reduction, straight line, histogram, segmentation, urban area.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES APPENDICES

LISTE DES SIGLES

3D	Three-Dimensional
ASM	Angular Second Moment
AVIRIS	Airborne Visible Infrared Imaging Spectrometer
CLH	Class Label Histogram
ICA	Independent Component Analysis
GLCM	Grey-Level Co-occurrence Matrix
GIS	Geographic Information Systems
HDV	High Dimensional Vector
HOG	Histogram of Gradients Orientation
HOM	Histogram of Gradient Magnitude
LLA	Level Line Angle
LLE	Locally Linear Embedding
LSD	Line Segment Detector
MDS	Multidimensional Scaling
NFA	Number of Falsa Alarms
PCA	Principal Component Analysis
PP	Projection-Pursuit
PRI	Probabilistic Rand Index

CHAPTER 1

INTRODUCTION

1.1 Motivation

In recent decades, urban detection and mapping from hyperspectral imagery has become important and crucial for several military and civilian purposes. The main application is Geographic Information Systems (GIS) update, which enables the study and planning of urban growth and sprawl. GIS are useful in order to explore suitable strategies for sustainable urban environmental development and to do periodic urban development planning. In addition, it is a real source of information in diverse fields such as geography, cartography (it can greatly help local agencies to update land maps and draw city plans), surveillance, city planning, and so on.

Very high resolution aerial and satellite images provide valuable information for this purpose. Unfortunately, these images cover very large areas and these (urban) areas are also dynamic environments. Therefore, the manual and periodic inspection of these satellite images is very hard and prone to errors. In this study, we propose an automatic and reliable segmentation method for detecting urban areas.

1.2 Related Background & Proposed Approach

A number of methods have been proposed and studied in the last decades to solve the difficult problem of satellite or aerial image segmentation and especially the urban area detection problem with unsupervised techniques. Most of these methods use a preliminary texture feature extraction step whose goal is to characterize each meaningful textured region (urban or not) with statistical (or textural, geometrical, morphological, fractal, etc.) image features which are then either characterized by their (parametric or non-parametric) distribution, or simply gathered

in a D-dimensional (feature) vector. This feature vector is then exploited in a subsequent clustering or classification procedure. In this classical strategy, years of research in segmentation have demonstrated that significant improvements on the final segmentation results may be achieved either by using notably more sophisticated feature selection procedures, or more elaborate clustering techniques.

For example, Li and Narayanan [LN04] have proposed to detect urban area based on Gabor wavelet and the support vector machine (SVM). Karathanassi *et al.* [KIR00] have used building-density information to classify residential regions. In a subsequent study, Unsalan and Boyer [UB04a, UB04b, Uns06] have identified and used a set of structural features to classify urban regions in panchromatic satellite images. Zhong and Wang [ZW07] have extracted urban regions in greyscale satellite images using combination of statistical models and multilevel structural features.

Similar to the work of Zhong and Wang [ZW07], we have also exploited multilevel structural cues; more precisely, we have based these structural features on statistics computed from the gradient orientation and straight segment detection. We have noticed that this set of features is especially relevant to describe different aspects of urban areas, which are mainly composed of manufactured objects exhibiting regular or simple geometrical shapes. We have also combined these structural cues with some classical textural features based on grey-levels, gradient magnitude or grey-level co-occurence matrix statistical parameters. Nevertheless, in this project, and in contrast to the work of Zhong and Wang, we have decided to directly use hyperspectral data as raw data instead of panchromatic satellite or aerial images.

A hyperspectral image is a kind of uncompressed image which is obtained directly from multispectral remote sensors such as the Landsat Thematic Mapper and SPOT XS. These hyperspectral imaging sensors, which have revolutionalised the field of remote sensing by combining the science of spectroscopy with that of imaging, have the ability to simultaneously collect the same image scene on many bands of the light spectrum, in dozens or hundreds of narrow, adjacent spectral

bands (see Fig. 1.1). The resulting 3D image, or hyperspectral data-cube, makes it possible to derive, for each pixel, a continuous and unique reflectance spectrum which is of great importance, for example, in geology and geophysics for identifying terrestrial surface materials such as particular mineral deposits or types of vegetation.

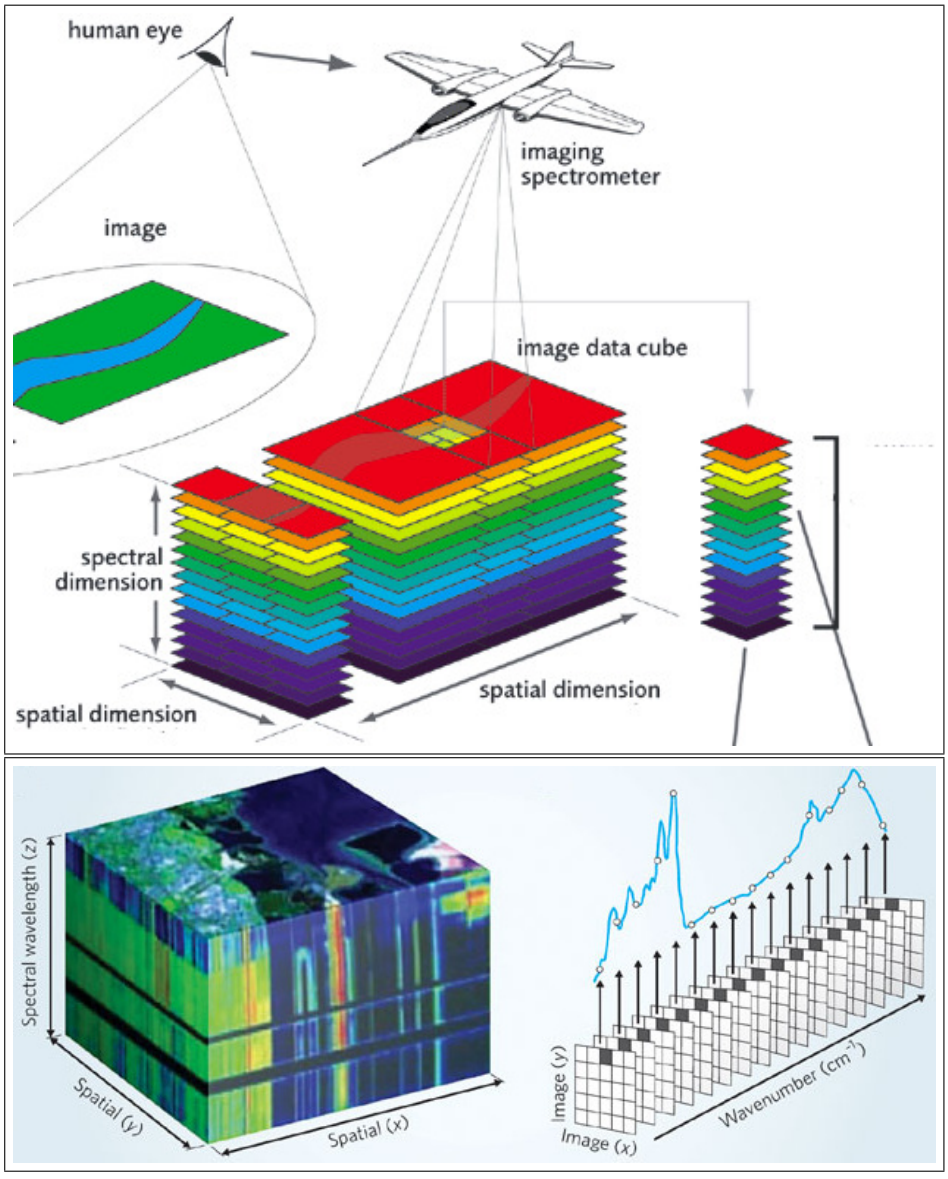


Figure 1.1: Hyperspectral image [Nat, Hyp]

Hyperspectral images provide the potential for more accurate and detailed information than common aerial or panchromatic satellite images. However, algorithms that calculate and analyze these raw high-dimensional data have a very high time complexity and are computationally expensive. In order to decrease computational complexity without restricting too much the inherent accuracy of these images, we have to compress the hyperspectral data cube in an informative three-color composite image, minimizing as much as possible the loss of information of the spectral content. To attain this goal, a non-linear dimensionality reduction model [Mig12], based on two complementary but contradictory criteria of good visualization, namely accuracy and contrast, is applied to the color display of each hyperspectral image. More precisely, the dimensionality reduction step used in this study achieves a trade-off between the fidelity to the un-reduced (raw) spectral data (accuracy) and some expected properties, such as the contrast, of the final color image. This subsequently allows *good separability* of each observed existing material in the final visualized color image.

Another contribution of this work is to consider a different approach for the classification procedure. Instead of simply gathering together in a high-dimensional (feature) vector all the the textural and structural features, we decide, in the final third step, to classify each individual feature by a simple K-means clustering procedure and to combine these multiple low-cost, weak and roughly estimated image segmentation results with a fusion model of segmentation maps such as the one proposed in [Mig08, Mig10a].

The idea behind a fusion approach is that it is very difficult to find a segmentation algorithm and/or selected features which could perfectly segment all hyperspectral images into two homogeneous regions such as urban and non-urban areas. On the other hand, it is logical to think that we could gain from combining the strengths and features of multiple segmentation maps which, individually, might produce some poor segments (*i.e.*, a poor segmentation result for some sub-parts

of the image) but for which there also often exist good segments. A clever merging of these segmentation results, with poor segments considered as noise (and good segments as reliable information), could produce a superior consensus segmentation than any of the individual input segmentations [Mig08, Mig10a]. This strategy also allows us to both reduce the computational complexity and to avoid the so-called "curse of dimensionality" when clustering high-dimensional data.

1.3 Project Structure

The overall procedure is illustrated in Figure 1.2.

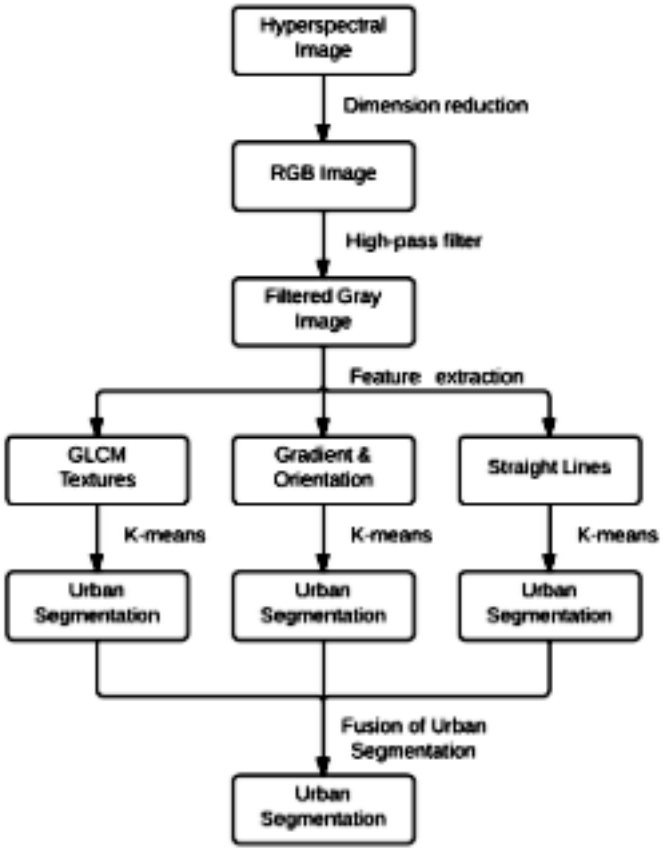


Figure 1.2: Structure of the project

The remainder of this master's thesis is organized as follows. Chapter 2 presents the broad theoretical and methodological background used in this study, especially the non-linear dimensionality reduction model which will be used, the Line Segment Detector (LSD) [vGJMR08] exploited in the structural feature extraction step, and finally the fusion model of segmentation maps. Chapter 3 presents the feature extraction step and the set of relevant structural and textural features whose goal is to describe the discriminant characteristics of the urban area, which are mainly composed of man-made objects with regular or simple geometrical shapes. Finally, chapter 4 shows various experiments and chapter 5 presents a discussion and conclusion.

CHAPTER 2

THEORETICAL BACKGROUND

2.1 Dimensionality Reduction Model

Hyperspectral images are generated from airborne sensors like the Landsat Thematic Mapper, SPOT XS, NASA's Airborne or AVIRIS etc. These imaging sensors allow us to collect the same image scene on many bands of the light spectrum. In this way, the resulting 3D image or hyperspectral data-cube[1] makes it possible to derive, for each pixel, a continuous and unique reflectance spectrum which is of great importance, for example, in geology and geophysics, for identifying earth's surface materials such as particular mineral deposits or types of vegetation. In practical applications, it is useful if this huge amount of high dimensional spectral data information is reduced to three dimensions. This allows us to quickly display this data cube into an informative color image (with red, green and blue channels) and to provide a quick overview of existing materials and their distribution in the image scene for further analysis [Mig10b].

To this end, dimensionality reduction methods based on linear projection methods such as Principal Component Analysis (PCA) [JR99] (and its numerous variants [SG84, Rog94, CD99]), Projection-Pursuit (PP) [JRACVR07], Independent Component Analysis (ICA) [DQWR03], linear projection-based strategy [JG05, JGC07] (and some similar strategies [KC10]), Multiresolution [RWK97] and wavelet decompositions [MLM95] have commonly been proposed in the literature to obtain the first three principal R, G and B image components to be finally visualized.

[1]The hyperspectral cubes used in this work are from the National Aeronautics and Space Administration Jet Propulsion Laboratory AVIRIS system [VGC+93], which captures 224 spectral bands, ranging from $400 - 2500$ nm with a 1995 Airborne Visible/Infrared Imaging Spectrometer (AVIRIS). The image size is usually around 512×614 pixels. We have used the reflectance data which is atmospherically corrected to compensate for absorption and the spectrum of the sun. The AVIRIS data are generously available for download on-line at http address HTTP://AVIRIS.JPL.NASA.GOV/HTML/AVIRIS.FREEDATA.HTML

However, it should be noted that the use of ICA or more generally, the use of a mutual information criterion [TA05], is based on the assumption of mutually independent sources, which is not really the case of hyperspectral data [ND05]. In order to overcome the main limitation of these above-mentioned linear projection methods (reviewed in [DRCM08]) which usually do not consider the inherent non-linear characteristics of the hyperspectral data [BAF05], an alternative idea consists of exploiting a (more time-consuming) non-linear dimensionality reduction method such as the locally linear embedding (LLE) method [MSB07] or the Multidimensional Scaling (MDS) based method [Mig08, CRHW09, Mig10b] and preserving the set of pairwise Euclidean spectral distances in the perceptual LAB space.

Obviously, every representation of a high dimensional space with only three dimensions (R, G and B channels) induces simplification and loss of information and should be done according to the most appropriate criterion. In this work, we rely on the non-linear dimensionality reduction model proposed in [Mig10b]. This dimensionality reduction model is based on a bi-criteria global optimization approach which is derived from two well-known and contradictory criteria of good visualization, useful in any multidimensional imagery color display, namely; accuracy, with the *preservation of spectral distance* criterion and contrast, guaranteeing that colors are well distinguished or concretely allowing the *good separability* of each observed existing material in the final visualized color image. In fact, the preservation of the spectral distance criterion alone does not guarantee that colors can be well distinguished. The final image may have low contrast, and consequently high contrast is necessary to get a clear visualization. This is attained, however, at the cost of information loss. We will briefly recall this model. For further details, the reader should refer to [Mig10b].

As already said, the main goal of the dimensionality reduction model is to preserve, as a first criterion, the distance between the spectral vector of each pair of pixels and their final perceptual color distance in the final displayed color image. The second criterion is related to the notion of contrast or the *separability of features*

and is by definition expressed by the average distance between each pair of pixels in LAB space (which is a perceptually uniform color space, *i.e.*, the same amount in a color value produces a change of about the same visual importance). To this end, let $\mathbf{u}$ be the three-dimensional vector ($\mathbf{u} = (L\ A\ B)^t$) corresponding to the three L, A, B color bands of the final image to be displayed. The dimensionality reduction model related to this bi-criteria color display model seeks to find $\hat{\mathbf{u}} = (\hat{L}\ \hat{A}\ \hat{B})^t$ which minimizes the following bi-criteria objective function:

$$\hat{\mathbf{u}} = \arg\min_{\mathbf{u}} \sum_{s,t_{s\neq t}} \left\{ \left(\beta_{s,t} - \left\| \phi(\mathbf{u}_s) - \phi(\mathbf{u}_t) \right\|_2 \right)^2 - \gamma \left\| \phi(\mathbf{u}_s) - \phi(\mathbf{u}_t) \right\|_2^2 \right\} \quad (2.1)$$

where the summation $\sum_{s,t_{s\neq t}}$ is over all the pairs of sites existing in the final image to be displayed, and $\beta_{s,t}$ denotes the Euclidean distance associated between two spectral vectors, associated with the pair of sites at spatial locations s and t. The first term is related to the preservation of the spectral distance criterion, and the second corresponds to the contrast criterion. Here, γ is the value controlling the contribution of these two criteria, and $\phi(\hat{\mathbf{u}}_\mathbf{s} = (\hat{L}_s\ \hat{A}_s\ \hat{B}_s)^t)$ is the function that takes into account the possible saturation effect, for each pixel, in the finally displayed RGB color space. More precisely, this function realizes the three following operations. First, it converts the LAB color values into RGB color values. Second, it ensures that all converted pixels are inside the RGB color space by setting negative pixel values to 0 and those that are greater than 255 to 255. Third, it converts back the RGB color values into the LAB color space. This function enables the contradiction existing between these two criteria to be taken into account. A high contrast (second criterion) in the perceptual LAB color space (see 2.1) can be obtained at the cost of numerous saturated pixels which can no longer satisfy the accuracy criterion of the resulting color mapping (first criterion).

In this context, the dimensionality reduction model is cast as a global optimization problem of a complex (non-convex) cost function over the LAB color value space. In order to find a particular configuration of $\hat{\mathbf{u}}$ that efficiently minimizes this complex energy function, we use the three-step optimization strategy

described in [Mig10b], namely:

1. First Step

   First, we ignore both the contrast criterion (*i.e.*, we take $\gamma = 0.0$) and the saturation effect (we take $\phi(\mathbf{u}) = \mathbf{u}$) and only keep the distance information of spectral image. In this context, the objective function to be minimized is equivalent to the loss function also used in multidimensional scaling methods for the estimation of a low-dimensional mapping from high-dimensional data:

   $$\hat{\mathbf{u}} \;=\; \arg\min_{\mathbf{u}} \sum_{s,t_{s\neq t}} \Big(\beta_{s,t} - \big\|\mathbf{u}_s - \mathbf{u}_t\big\|_2\Big)^2. \tag{2.2}$$

   Further, we use the FastMap algorithm [FL95] (or its variants [Pla05]) either directly for estimating a three dimensional (3D) embedding, *i.e.*, for the 3D mapping $\mathbf{u} = (L\,A\,B)^t$ (see Eq. (2.2)) or for the separate estimation of L and A and B (see Eq. (2.3)). Each (one-dimensional) mapping being defined (in our case) for the three equal-sized subsets covering the overall available wavelengths of the original hyperspectral image, *i.e.*,

   $$\left\{\begin{array}{rcl} L &=& \arg\min \sum_{s,t_{s\neq t}} \Big(\beta_{s,t}^{[1:\frac{1}{3}K]} - |L_s - L_t|\Big)^2 \\ A &=& \arg\min \sum_{s,t_{s\neq t}} \Big(\beta_{s,t}^{]\frac{1}{3}K:\frac{2}{3}K]} - |A_s - A_t|\Big)^2 \\ B &=& \arg\min \sum_{s,t_{s\neq t}} \Big(\beta_{s,t}^{]\frac{2}{3}K:K]} - |B_s - B_t|\Big)^2 \end{array}\right. \tag{2.3}$$

   where $\beta_{s,t}^{[k_0:k_1]}$ denotes the Euclidean distance between the spectral bands k_0 and k_1 and L_s, A_s, B_s are respectively the L, B and A components at the site (or pixel) s. Experimental results in [Mig10b] show that minimizing Eq. (2.3) gives a better initialization step to our final optimization problem (this is certainly due to the fact that the FastMap is all the more efficient when the embedding is achieved at very low dimensions [Pla05]). In order to now ensure that the LAB color values of the 3D mapping $\mathbf{u}$ are mostly not saturated in the RGB space (*i.e.*, a very small minority of pixels are outside

the RGB color space), we use a simple linear stretching of the L, A, B color values such as $L \in [0 : 100]$, and A, B have a maximal amplitude of 100 (with a zero mean). Mathematically, this helps ensure that $\phi(\mathbf{u}_s) = \mathbf{u}_s$ for all sites s and concretely that there are no saturated pixels which could alter the *preservation of spectral distance*. This linear stretching allows us to find the color mapping that minimizes the following objective function criterion

$$\hat{\mathbf{u}} \;=\; \arg\min_{\mathbf{u}} \sum_{s,t_{s\neq t}} \left(\beta_{s,t}^{\text{scaled}} - \left\| \phi(\mathbf{u}_s) - \phi(\mathbf{u}_t) \right\|_2 \right)^2 \tag{2.4}$$

where $\phi(\mathbf{u}_s) \approx \mathbf{u}_s$, due to our suitable range of LAB color values and $\beta_{s,t}^{\text{scaled}} = \rho\, \beta_{s,t}$. Here, ρ is a scaling factor ensuring that, after the linear stretching, the pairwise distances of the 3D mapping $\mathbf{u}$ and the distance of the pairwise spectral vectors are still preserved. This scaling factor is defined by

$$\rho = \frac{\sum_{s,t} d_{s,t}^{\text{stretch}}}{\sum_{s,t} d_{s,t}} \tag{2.5}$$

where $d_{s,t}$ and $d_{s,t}^{\text{stretch}}$ are respectively the set of pairwise Euclidean distances of the image before and after the linear stretching process.

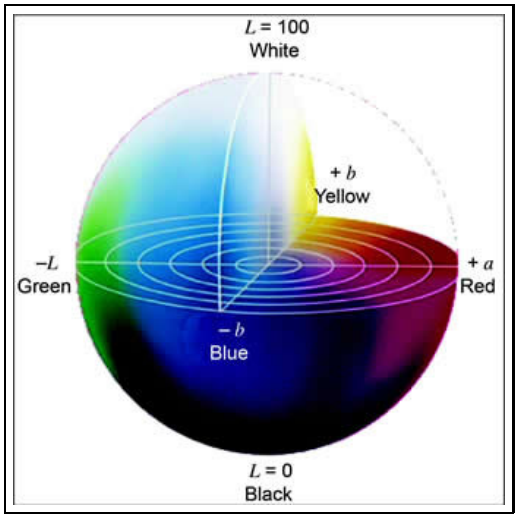


Figure 2.1: LAB perceptual color space [Son]

2. Second Step

   At this stage, we search for the maximal increase in the contrast of the image by multiplying the LAB color values by a spatially variant positive scaling factor (at a price that some pixels will become saturated in the RGB space) or by estimating the spatially invariant scaling factor k for which

$$\hat{k} = \arg\min_k \sum_{s,t_{s\neq t}} \left\{ \left(k\beta_{s,t}^{\text{scaled}} - \left\| \phi(k\mathbf{u}_s) - \phi(k\mathbf{u}_t) \right\|_2 \right)^2 - \gamma \left\| \phi(k\mathbf{u}_s) - \phi(k\mathbf{u}_t) \right\|_2^2 \right\} \tag{2.6}$$

   In order to do this estimation, we use a simple local discrete grid search, for the parameter k in a suitable range ($k \in [1.0-3.0]$ with a fixed step size set to 0.1). In order to decrease the computational time, we consider that each site is connected only with its four nearest neighbors and four equally spaced other pixels located within a square neighbourhood window of fixed size $N_s = 61$ pixels (empirically chosen based on trial and error) centered around the pixel (see Fig. 2.2). Since the above-mentioned process can saturate some pixels in

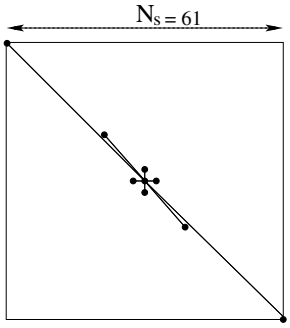


Figure 2.2: Spatial neighborhood

   the RGB space and that may alter the optimal value of $k\beta_{s,t}^{\text{scaled}}$, we refine the estimation of this parameter with a least-squares estimation. More precisely, $k\beta_{s,t}^{\text{scaled}}$ is corrected by the factor [Pla05]

$$\rho = \frac{\sum_{s,t} \hat{\beta}_{s,t}^2 / \beta_{s,t}^2}{\sum_{s,t} \hat{\beta}_{s,t} / \beta_{s,t}} \tag{2.7}$$

where $\beta_{s,t}$ is the true (unembedded) distance, using the set of pairwise spectral vectors of the hyperspectral image and $\hat{\beta}_{s,t}$ is the estimated (in the embedded space) distance, using the set of pixel pairs in the stretched $\mathbf{u}$ mapping.

3. Third Step

   At this stage, we are close to the solution of the optimization problem expressed in Eq. (2.1). We further refine the last obtained estimation (Eq. (2.6)) by a local exploration around the current solution using the Metropolis algorithm [MRR+53] and a low radius of exploration (see Algorithm 1).

Algorithm 1
Local exploration with Metropolis

E	Energy function to be minimized
T_l	Temperature at Iteration step l
a	Cooling schedule parameter

r	Radius of exploration, real $\in]0,1]$
T_0, T_f	Initial and final temperature
$L^S_{\max}$	Maximal number of iterations

1. Initialization $a \leftarrow \left(\frac{T_f}{T_0}\right)^{\frac{1}{L^S_{\max}}}$

2. Local Exploration

while $l < L^S_{max}$ **do**

  for *each pixel and each color channel with value* x_s *at site* s **do**

    • Compute $\Delta_{\text{Energy}} = E(y_s) - E(x_s)$ with $y_s \in [x_s - r : x_s + r] \quad (x_s, y_s \in [0.0\,{:}\,1.0])$

    • **if** ($\Delta_{\text{Energy}} < 0$) Replace x_s by y_s

    • **else** Replace x_s by y_s with probability $\triangleright \quad \exp\left(-\frac{\Delta_{\text{Energy}}}{T_l}\right)$

  $l \leftarrow l + 1$ and $T_l \leftarrow T_0\, a^l$

Figure 2.3: Algorithm 1. Local exploration with Metropolis [Mig12]

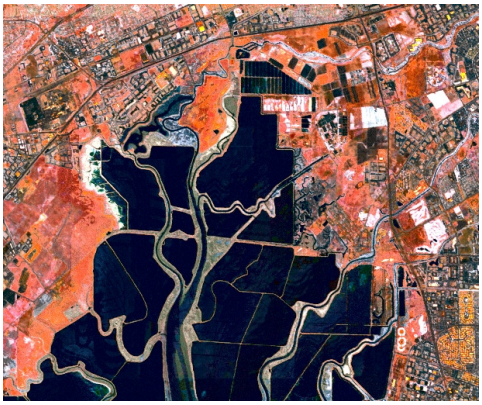
Finally, we get a LAB color space image as a feasible solution for our bi-criteria optimization problem, and we need to convert it into RGB color space for the purpose of visualization. We notice (see Fig. 2.4) that the different regions can be easily identified and very fine details are visible without artifacts, such as noise or blurring artifacts due to the reduction/compression model or spectral ringing artifacts, glints and anomalous fluorescence hot spots.

For a set of increasing discretized values of γ starting from $\gamma = 0$ (thus increasing the contrast of the image), we can then compute the value of the *preservation of spectral distance* based measure (with a correlation score measuring the agreement between each pairwise spectral vector in the full and reduced dimension; a correlation value equal to 1 means that there is no loss of information), and stop when this measure starts to decrease significantly or when this measure is below a fixed value of accuracy. For example, for the bottom-right image of Fig. 2.4 (with $\gamma = 0.4$), the contrast enhancement is achieved, relative to the bottom-left image ($\gamma = 0.3$), to the detriment of 6.3 per cent (*i.e.*, $0.95761 - 0.895 \approx 0.063$), of pairs of pixels which can no longer satisfy the agreement between the distance of spectrums (associated to each pair of pixels) and their perceptual color distance in the final fused image to be displayed (*i.e.*, 6.3 percent loss of accuracy for this contrast improvement).

$\gamma = 0.1$, Correlation=0.95181

$\gamma = 0.2$, Correlation=0.96071

$\gamma = 0.3$, Correlation=0.95761



$\gamma = 0.4$, Correlation=0.895

Figure 2.4: Compression results

2.2 Multiple Feature Extraction Model

2.2.1 LSD Model

In this section, we introduce another important tool for this project. This tool, recently introduced in [vGJMR10], is an efficient linear-time line segment detector that gives sub-pixel accurate detection results for a controlled pre-determined number of false detections according to the *a contrario* principle (and without requiring parameter tuning).

In this work, we have used this detector to describe the geometric content (*via* some structural features) and the straight lines (or line segments) and regular structures present in hyperspectral images. More precisely, we have noticed that the spatial arrangement of these straight lines, over relatively small neighborhoods (*via* the computation of some basic statistics) is especially relevant to discriminate a urban area that is mainly composed of manufactured objects exhibiting regular or simple geometrical shapes. Indeed, most human-made objects or manufactured structures, like buildings, bridges, roads etc. are made of flat surfaces and can be well described by a finite number of straight lines which are often either parallel or perpendicular.

In contrast to classic edge detectors, this LSD algorithm defines a line segment as a rectangular region whose points share roughly the same image gradient angle. This algorithm is very efficient for the detection of line segments based on the empirical discovery made by Burns *et al.* [BHR86] showing that connected (rectangular) regions with common orientation would almost always coincide with straight edges.

2.2.1.1 Line Support Region

The LSD algorithm starts by computing the level-line angle at each pixel to produce a level-line field, *i.e.*, a unit vector field such that all vectors are tangent to the level line going through their base point. Then, this field is segmented into

connected regions of pixels that share the same level-line angle up to a certain tolerance. The whole algorithm can be described in three steps (see also Fig. 2.5):

1. Partition the image into line-support regions by grouping connected pixels that share similar gradient angles up to a certain tolerance.

2. Approximate each line support region into the best line segment.

3. Validate or not each line segment based on the information in the line-support region.

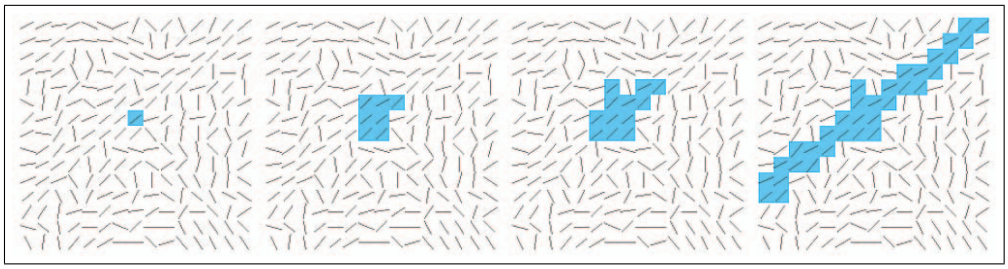


Figure 2.5: Growing process of a region [vGJMR10]

The gradient orientation field in Fig. 2.5 is represented by dashes. Marked pixels are the ones forming the region (from left to right: first, second, and third iterations, and final result). More precisely, each region starts with one pixel and the initial region angle is set to be the level line angle (gradient angle) at that pixel. Then, the pixels adjacent to the region are tested; the ones with gradient orientation equal or similar to the region angle is added to this region. At each iteration, the region angle is updated until no new adjacent pixel is added to that region. The region angle is defined by

$$\arctan = \frac{\sum_i \sin(ang_i)}{\sum_i \cos(ang_i)}. \tag{2.8}$$

The pseudo-code (RegionGrow, Fig. 2.6) gives more details. Seed pixels with larger gradient magnitude are tested first as they are more likely to belong to straight edges. When a pixel is added to a region, it is marked used and never visited again.

Algorithm 1. REGIONGROW

input: An image I; a starting pixel (x, y); an angle tolerance τ; an image *Status* where pixels used by other regions are marked.

output: A list *Region* of pixels.

```
1  Region ← (x, y);
2  θ_region ← LevelLineAngle(x, y);
3  S_x ← cos(θ_region);
4  S_y ← sin(θ_region);
5  foreach pixel P in Region do
6     foreach P̄ neighbor of P and Status(P̄) ≠ Used do
7        if Diff(LevelLineAngle(P̄), θ_region ) < τ then
8              Add P̄ to Region;
9              Status(P̄) ← Used;
10             S_x ← S_x + cos(LevelLineAngle(P̄));
11             S_y ← S_y + sin(LevelLineAngle(P̄));
12             θ_region ← arctan(S_y/S_x);
13       end
14    end
15 end
```

Figure 2.6: RegionGrow algorithm [vGJMR10]

2.2.1.2 Rectangle Approximation of Regions

A valid line-support region must be associated with a line segment. A line segment is actually a rectangle, determined by its two endpoints and its width. Its approximation is shown in Fig. 2.7. Line segments are characterized by a rectangle determined by its center point, angle of orientation, length, and width. In order to estimate these parameters, we first regard the region of pixels as a solid object and the gradient magnitude of each pixel is used as the *mass* of that point. Then the center of mass is used to select the center of rectangle. The angle is the main direction for the line segment (this procedure can lead to an erroneous line angle estimation when the background shows a slow intensity variation, see [vGJMR08]). Finally, the length and the width are set to the smallest values that cause the rectangle to cover the full line support region. The center of the rectangle (c_x, c_y)

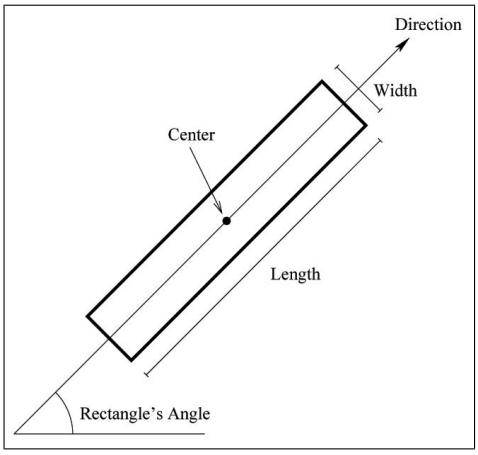


Figure 2.7: Rectangle approximation of regions [vGJMR10]

is set to

$$c_x = \frac{\sum_{j \in \text{Region}} G(j) \cdot x(j)}{\sum_{j \in \text{Region}} G(j)} \tag{2.9}$$

$$c_y = \frac{\sum_{j \in \text{Region}} G(j) \cdot y(j)}{\sum_{j \in \text{Region}} G(j)} \tag{2.10}$$

where $G(j)$ is the gradient magnitude of pixel j, j runs over the pixels in the region, the main rectangle's angle is set to the angle of the eigenvector associated with the smallest eigenvalue of the matrix:

$$M = \begin{pmatrix} m^{xx} & m^{xy} \\ m^{xy} & m^{yy} \end{pmatrix}$$

with

$$m^{xx} = \frac{\sum_{j \in \text{Region}} G(j) \cdot (x(j) - c_x)^2}{\sum_{j \in \text{Region}} G(j)} \tag{2.11}$$

$$m^{yy} = \frac{\sum_{j \in \text{Region}} G(j) \cdot (y(j) - c_y)^2}{\sum_{j \in \text{Region}} G(j)} \tag{2.12}$$

$$m^{xy} = \frac{\sum_{j \in \text{Region}} G(j) \cdot (y(j) - c_y)\,(x(j) - c_x)}{\sum_{j \in \text{Region}} G(j)} \tag{2.13}$$

Figure 2.8 shows an example of a rectangular approximation of a line-support region. (Left: Image. Middle: One of the line-support regions. Right: Rectangular approximation superposed on the line-support region).

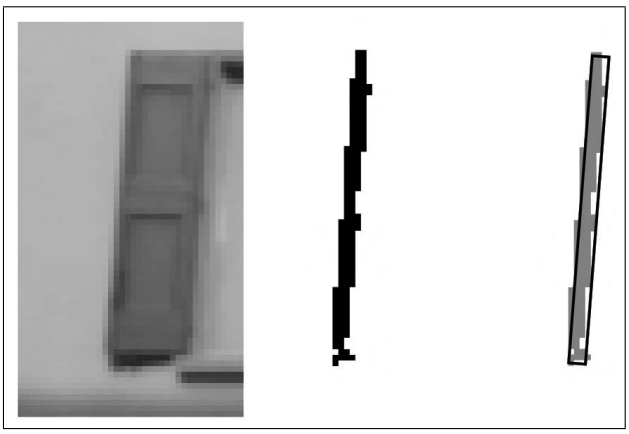


Figure 2.8: Example of rectangular approximation [vGJMR10]

2.2.1.3 Line Segment Validation

The line segment validation procedure is to test whether or not a line segment belongs to foreground, according to the theory of the contrario approach and the Helmholtz principle as proposed by Desolneux *et al.* [DMM00]. The so-called Helmholtz principle states that no detection (of straight lines) should be produced on an image of noise, in which all gradient angles are independent and uniformly distributed. Conversely, structured events (all gradient angles in the region correspond to the angle of the rectangle up to a certain tolerance) are defined as being rare in the *a contrario* model and have to be considered as non-meaningful straight lines. Thus, a valid line segment should have many or more aligned points (*i.e.*, gradient angles corresponding to the angle of the rectangle up to a certain tolerance). Given an image i and a rectangle r, we denote by $k(r, i)$ the number of aligned points and $n(r)$ the total number of pixels in r. Then, the expected number of events is given by

$$N_{\text{test}} \times P_{H_0}[k(r, I) \geq k(r, i)] \tag{2.14}$$

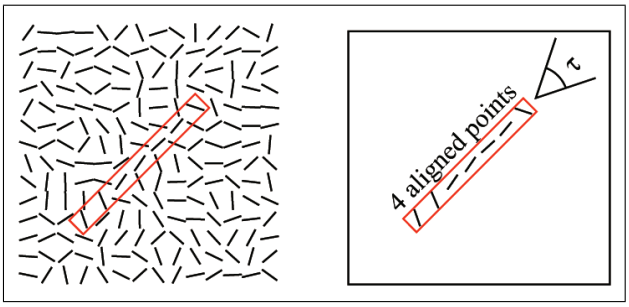


Figure 2.9: Example of aligned points [vGJMR10]

where the number of tests N_{test} is the total number of possible rectangles being considered, P_{H_0} is the probability of the background model H_0 , and I is a random image following H_0. The H_0 stochastic model fixes the distribution of the number of aligned points $k(r, I)$, which only depends on the distribution of the level line field associated with I. The background model H_0 satisfies the following properties:

1. $\{LLA(j)\}_{j \in \text{Pixel}}$ is composed of independent random variables;

2. $LLA(j)$ is uniformly distributed over $[0, 2\pi]$;

here, $LLA(j)$ is the level line angle at pixel j. Under hypothesis H_0, the probability that a pixel on a background model is an aligned point is $p = \tau/\pi$ and, as a consequence of the independence of the random variables $LLA(j)$, $k(r, I)$ follows a binomial distribution. Thus the probability term:

$$P_{H_0}[k(r, I) \geq k(r, i)] \tag{2.15}$$

is given by

$$P_{H_0}[k(r, I) \geq k(r, i)] = B(n(r), k(r, i), p) \tag{2.16}$$

where $B(n, k, p)$ is the tail of the binomial distribution:

$$B(n, k, p) = \sum_{j=k}^{n} \binom{n}{j} p^j (1-p)^{(n-j)}. \tag{2.17}$$

The number of tests N_{test} corresponds to the total number of rectangles. The exhaustive choice is to take all the rectangles starting and ending at image pixels. In an $N \times M$ image this gives $NM \times NM$ different rectangles, $\sqrt{MN}$ different width values are considered for each one. The total number of possible rectangles

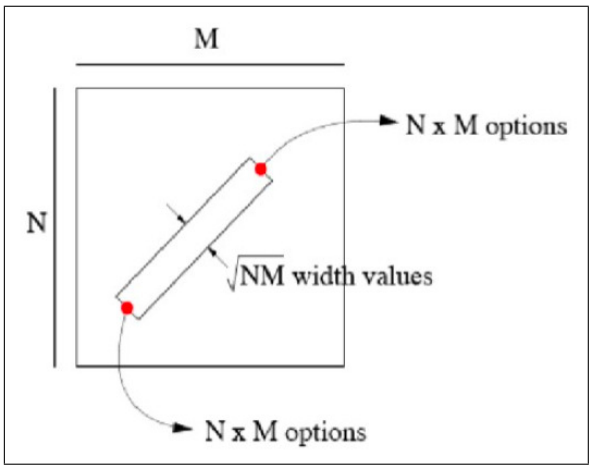


Figure 2.10: Possible detected rectangle [vGJMR10]

in a picture of $N \times M$ pixels is $NM^{5/2}$. The value p is initially set to τ/π, but other values are also tested to cover the relevant range of values. Finally γ is the number of different p values. Each rectangle is tested with each p value. The final number of tests is $\gamma\, NM^{5/2}$. Finally, we could get the Number of False Alarms (NFA):

$$\mathrm{NFA}(r,i) = \gamma\, MN^{5/2}\, B(n(r), k(r,i), p) \tag{2.18}$$

This corresponds to the expected number of rectangles which have a sufficient number of aligned points to be as rare as r under H_0. When the NFA is large, this means that such an event is expected to be non-meaningful. On the other hand, when the NFA value is small, the event is acceptable. A threshold ϵ is set: we select the valid line segment which satisfies the inequality: $\mathrm{NFA}(r,i) \leq \epsilon$.

2.2.1.4 The Complete LSD Algorithm

Algorithm 2 shows a pseudo-code for the complete algorithm. The region growing process starts with the pixel that has the largest gradient magnitude and ends with the smallest one (because the straight edges are more likely to be formed by the pixels with larger gradient value). Line 5: RegionGrow, is used to obtain a line-support region. Line 6: RectApprox, gives a rectangle approximation of the region. Line 7: NFA is finally used to validate or not each line-segment, and we recall that these methods are previously described in Sections 2.2.1.1, 2.2.1.2, 2.2.1.3.

Algorithm 2. LSD: LINE SEGMENT DETECTOR
input: An image I, parameters ρ, τ and ε.
output: A list *out* of rectangles.

```
1  (LLAngles, GradMod, OrderedListPixels) ← Grad(I, ρ);
2  Status(allpixels) ← NotUsed;
3  foreach pixel P in OrderedListPixels do
4     if Status(P) = NotUsed then
5        region ← RegionGrow(P, τ, Status);
6        rect ← RectApprox(region);
7        nfa ← NFA(rect);
8        nfa ← ImproveRect(rect);
9        if nfa < ε then
10          Add rect to out;
11          Status(region) ← Used;
12       else
13          Status(region) ← NotIni;
14       end
15    end
16 end
```

Figure 2.11: LSD complete algorithm [vGJMR10]

2.2.1.5 Internal Parameters

There are three parameters involved in LSD: ρ, τ, and ϵ. The parameter ρ is a threshold on the gradient magnitude: pixels with small gradient are not considered. Desolneux *et al.* [DMM00] showed that gray level quantification produces errors in the gradient orientation angle. This error is negligible when gradient magnitude is large, but can be significant for a small gradient magnitude. We define

$$| \text{ angle error } | \leq \arcsin\Big(\frac{q}{\Delta n}\Big) \tag{2.19}$$

where q is the noise added in the image and Δn is the magnitude error between an ideal image and a real image. We have to reject pixels where the angle error is larger than angle tolerance τ used in the RegionGrow algorithm. That is, we impose $|angle\ error| \leq \tau$ and we get $\tau = \frac{q}{sin(\tau)}$. In this case we set $q = 5$ and τ is 30 degree after several tests.

2.2.1.6 Structural Features From LSD

In our study, urban and rural regions can be discriminated from several discriminant structural features based on a previous straight line detection step. In this work, we have considered the following structural features:

- Average line length
- Entropy of line length
- Average line contrast
- Entropy of line contrast

The reason we choose these four features to describe a region is based on the fact that wilderness and rural areas produce more random line length and lower line contrast. Consequently, these features allow us to get a good distinction between urban and rural area. This will be made more explicit in chapter 3.

2.2.2 GLCM Model

The Grey-Level Co-occurrence Matrix (GLCM), introduced by Haralick [HSD73] in the 1970s, has proven to be an effective texture descriptor and has since been used in numerous fields to quantify the textural pattern within an image (given a specified analysis window). Indeed, various textural parameters calculated from the grey level co-occurrence matrix help understand the details about the overall image content. It is defined as "a two dimensional histogram of gray levels for each pair of pixels, which are separated by a fixed spatial relationship."The GLCM of an image is defined by its radius δ and orientation θ. For example, consider a 4×4 matrix with four grey-tone values 0 through 3. A generalized GLCM for such an image is shown below where number of (i, j) stands for number of times that grey values i and j have been neighbors satisfying the condition stated.

0	0	1	1
0	0	1	1
0	2	2	2
2	2	3	3

Table 2.1: Grey value

Grey Tone	0	1	2	3
0	$\sharp(0,0)$	$\sharp(0,1)$	$\sharp(0,2)$	$\sharp(0,3)$
1	$\sharp(1,0)$	$\sharp(1,1)$	$\sharp(1,2)$	$\sharp(1,3)$
2	$\sharp(2,0)$	$\sharp(2,1)$	$\sharp(2,2)$	$\sharp(2,3)$
3	$\sharp(3,0)$	$\sharp(3,1)$	$\sharp(3,2)$	$\sharp(3,3)$

Table 2.2: General form of GLCM

The four GLCM for angles equal to 0^0, 45^0, 90^0 and 135^0 and radius equal to 1 are thus given by:

4	2	1	0
2	4	0	0
1	0	6	1
0	0	1	2

Table 2.3: GLCM ($\delta=1$, $\theta=0^0$)

6	0	2	0
0	4	2	0
2	2	2	2
0	0	2	0

Table 2.4: GLCM ($\delta=1$, $\theta=90^0$)

After making the GLCM frequency tables, we calculate the probability of each pair of adjacent pixels for each direction. For example, for Table 2.1 ($\delta = 1, \theta = 0^0$), the probability of $P(0,0) = 4/(4+2+1+2+4+1+6+1+1+2) = 1/6 = 0.166$

4	1	0	0
1	2	2	0
0	2	4	1
0	0	1	0

Table 2.5: GLCM ($\delta=1$, $\theta=45^0$)

2	1	3	0
1	2	1	0
3	1	0	2
0	0	2	0

Table 2.6: GLCM ($\delta=1$, $\theta=135^0$)

0.166	0.083	0.042	0
0.083	0.166	0	0
0.042	0	0.249	0.042
0	0	0.042	0.083

Table 2.7: Statistics example of GLCM

and then we have probability Table 2.7. According to this probability Table 2.7, it is now straightforward to compute, for example, the so-called contrast feature for ($\delta = 1$,$\theta = 0^0$):

$$\text{Contrast} = \sum_{i,j=0}^{N-1} P(i,j)(i-j)^2 \tag{2.20}$$

where N is the total number of grey level in windows. More specifically,

$$\begin{aligned}
\text{Contrast} &= 0.166\times(0-0)^2 + 0.083\times(0-1)^2 + 0.042\times(0-2)^2 + 0\times(0-3)^2 \\
&+ 0.083\times(1-0)^2 + 0.166\times(1-1)^2 + 0\times(1-2)^2 + 0\times(1-3)^2 \\
&+ 0.042\times(2-0)^2 + 0\times(2-1)^2 + 0.250\times(2-2)^2 + 0.042\times(2-3)^2 \\
&+ 0\times(3-0)^2 + 0\times(3-1)^2 + 0.042\times(3-2)^2 + 0.083\times(3-3)^2 \\
&\approx 0.586
\end{aligned}$$

There will be some other textural measures which will be based on these GLCM, such as entropy, energy, correlation, and which will be considered for this project as textural features. This will be detailed in the following chapter.

2.2.3 Gradient Magnitude and Orientation Model

Histogram of Gradients Orientation (HOG) and Magnitude (HOM) are feature descriptors often used in computer vision and image processing for the purpose

of object detection. The technique counts occurrences of gradient orientation and magnitude in a local window of an image. For example, Dalal and Triggs [DT05] have successfully used HOG descriptors to solve the problem of pedestrian detection in static images.

Similar to the method of Dalal and Triggs [DT05], we can also exploit HOG and HOM in order to efficiently discriminate between man-made and natural structures by extracting textural features from the histogram estimated on each local (overlapping) sliding $n \times n$ window centered around each pixel to be classified. For example, if a window contains urban texture (e.g., with buildings) we can likely obtain a histogram of gradient orientation with two main peaks because of the common sense that buildings are described by straight lines which are often either parallel or perpendicular and since most of the buildings are oriented perpendicularly to the streets which are also often parallel (between them).

2.3 Fusion of Segmentation Maps

Fusion of Segmentation Map [Mig08] is our final tool in this project. Each individual (textural and structural) feature will be clustered independently of the others with a simple K-means clustering procedure, and a final fusion procedure will aim to combine these segmentation maps in order to provide a more reliable and accurate segmentation result. The advantage of a fusion approach, in our case, is fourfold:

- First, it is very difficult to find a segmentation algorithm and/or selected features which could perfectly segment all kinds of hyperspectral images into two homogeneous regions such as urban and non-urban areas. On the other hand, it is logical to think that we could gain from combining the strengths and features of multiple segmentation maps which, individually, might produce some poor segments (*i.e.*, poor segmentation result for some sub-parts of the image) but for which there also often exist good segments. A clever merging of these segmentation results, with poor segments considered as noise (and

good segments as reliable information), could produce a superior consensus segmentation than any of the individual input segmentations [Mig08,Mig10a].

- This strategy also allows us to both reduce the computational complexity and to avoid the so-called "curse of dimensionality" when clustering high-dimensional data (as would be the case by gathering all the considered textural and structural features into a unique and high-dimensional feature vector). Of course, a dimensionality model may be used but every dimensionality reduction model implies a loss of information (since, we are not ensured than the global feature vector lies on an embedded low-dimensional linear or a non-linear manifold within the higher-dimensional space).

- This strategy avoids the rescaling problem which occurs when different features with different units are blended together. Indeed, it prevents that a similarity measure (used to evaluate the distance between feature vectors during the clustering procedure) (wrongly) gives a overwhelming importance to a feature having a larger unit range, and conversely.

- Finally, this fusion strategy will allow simply to take into account the important features of a (textured) natural image, *i.e.*, the inherent spatial dependencies between spatial neighbouring sites (which are likely to belong to the same region).

2.3.1 Label Class Histogram Based Fusion Model

The key idea behind the fusion procedure proposed in [Mig08] is to simply consider, for each pixel in a fixed window, the local histogram of the class labels of every segmentation map to be fused, For example, in order to fuse N segmentations with M classes into a segmentation with P classes, for each pixel we first build N histograms (N segmentation maps) with M bins (M classes) computed on each local (overlapping) sliding small (squared) window centered around each pixel to be classified, in order to get a $N \times M$-dimensional feature vector for every pixel

(then normalized to sum to one). We finally use a K-means clustering method to obtain a segmentation into P classes.

Consequently, in this fusion procedure, two sites for which the local-class-label histogram (*i.e.*, the mixture of textons in the different color spaces given by the bins histogram) are not too far away from each other will be in the same class in the resulting fused segmentation. Inversely, two sites associated to different local-class-label histograms will likely belong to different classes in the final segmentation.

An example of fusion results is given in Fig. 2.12 (from top to bottom and left to right; input natural image and six segmentation results obtained by a K-means clustering on an input image expressed in the RGB, HSV, YIQ, XYZ, LAB and LUV color spaces with a coarsely quantized color histogram as feature vector and final segmentation map resulting of the fusion of these six clusterings (bottom right)).

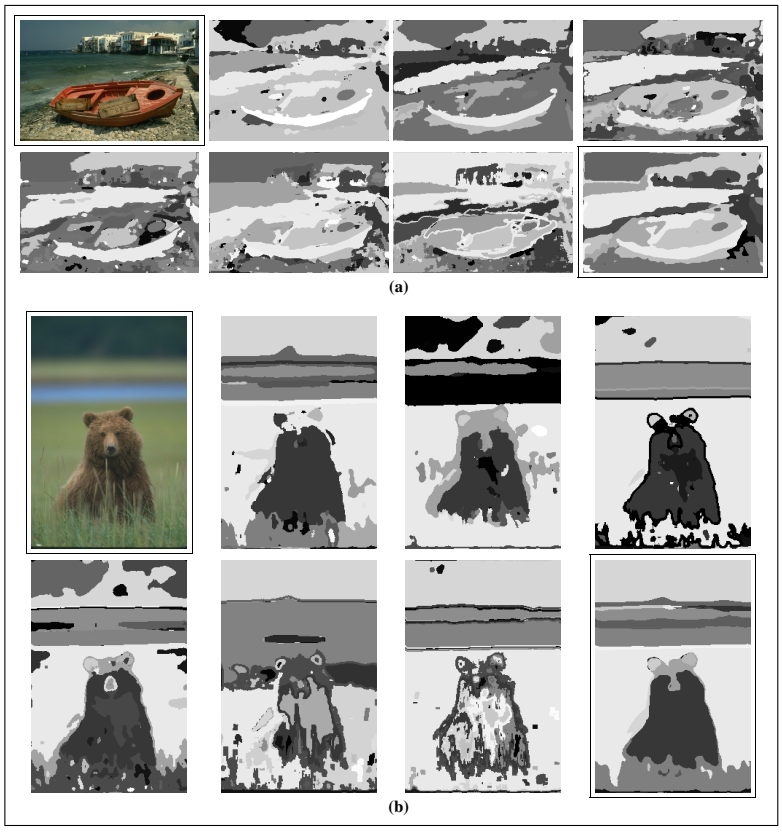


Figure 2.12: Example of fusion result on a natural image [Mig08]

2.3.2 PRI Based Fusion Model

This fusion of segmentation maps can also be achieved in the Probabilistic Rand Index (PRI) [Mig10b] sense, with a consensus (or loss) function encoding the set of constraints, in terms of pairs of pixel labels (identical or not) provided by each of the segmentations to be fused. The resulting optimization can then be solved with a multiscale Markovian approach combined with a coarse-to-fine optimization method (see [Mig10b] for further details). In this framework, the resulting fusion segmentation map $\hat{S}_{\text{fusion}}$ is defined by maximizing the following full-scale energy function:

$$\hat{S}_{\text{fusion-PRI}} = \arg\min_{S} \Big\{ \sum_{<i,j>} \delta(l_i, l_j) \big[1 - 2p_{ij}\big] \Big\} \tag{2.21}$$

where the set of $\{p_{ij}\}$ is computed with the empirical proportion estimator on the set of L segmentation S_k $(k \leq L)$ to be fused with

$$p_{ij} = \frac{1}{L} \sum_{k=1}^{k=L} \delta(l_i^{S_k}, l_j^{S_k}) \tag{2.22}$$

where δ is the delta Kronecker function and $l_i^{S_k}$ is the label of S_k at site i. For example, if a pair of sites $< i, j >$, over all the segmentations to be fused, have the same label, $p_{ij} = 1.0$. If a pair of sites $< i, j >$, over all the segmentations have different labels, $p_{ij} = 0.0$. Concretely, this term encodes the set of constraints, in terms of pairs of pixel labels (identical or not), provided by each of the L segmentations to be fused. The minimization of this term finds the resulting segmentation which also optimizes the PRI criterion. For more details concerning the minimization of this function, see [Mig10b].

CHAPTER 3

FEATURE EXTRACTION

3.1 Introduction

In remote sensing images, urban areas show obvious different textural characteristics from those of non-urban areas. An urban area is a topographical complex region where man-made constructions are the dominant features. These man-made structures show several characteristics in terms of various textural and structural features. In terms of structural features, for example, buildings usually show many straight-line segments, and their roofs are regular shapes and as already said, buildings are described by straight lines which are often either parallel or perpendicular with two main orientations. In terms of textural features, for example, residential buildings usually reflect more light than other objects do, thus the grey-levels in these areas are much brighter with a greater variance.

(a) (b)

Figure 3.1: Different resolution images

In addition, we have also to take the image resolution into consideration. In the mid-resolution images (10m per pixel), the urban areas usually show many short straight line segments produced by buildings and houses (Fig. 3.1(a)). But in the images with high resolution, (2m per pixel) [3.1(b)], the same buildings are often much larger and produce not only short line segments but also many long line segments.

We have implemented three types of features in this project: the textural features with respectively the GLCM and the gradient magnitude features; the local structural features with the gradient orientation features; and, the global structural features with the straight line based features. The gradient-magnitude and GLCM features are both grey-level based features: they describe grey-tones change tendency. While straight line and gradient-orientation features captured relatively strong or high-level structural information, they are both structure-based features.

We have computed the features by using different window scales according to the image resolution. The meaning of multiple scales is the same as introduced in the study in [KH03]. In this case, we set a small window size 4×4 around every pixel for computing the GLCM features, gradient magnitude and orientation features since image resolution is low, but for straight line segments we choose a slightly larger window size 12×12 in the purpose of getting more line segments in that window. The more line segments there are, the more accurate statistics we can get.

3.2 Gradient Magnitude Features

At first, we built the gradient magnitude histogram M for each pixel s, and normalized it by $\text{MAX}_s \in \{M_s\}$ (*i.e.*, $M_s \rightarrow M_s/\text{MAX}_s$). More precisely, for each image pixel s, a histogram $\{H(i)_s, i = 1, 2, \cdots, N\}$ was computed over the normalized gradient magnitudes contained in a window W_s with size $n_s \times m_s$ centered around s. The magnitude value of every pixel in the normalized gradient magnitude image lies between zero and one. We divided $[0, 1]$ into N equal intervals (or bins,

here $N = 20$). When the magnitude value falls into the interval $[i/N, (i+1)/N)$, the count of $H_s(i)$ increases by one. Using the histogram as an approximate probability $P(i)$ mass function of gradient magnitude, we can then compute the following features from this non-parametric gradient magnitude histogram [RK82]

- Magnitude Mean:

$$\mu_M = \sum_{i=0}^{N} P(i)\, i/N \tag{3.1}$$

  The mean tells us something about the general magnitude of the image. An urban region usually has a high mean value.

- Magnitude Variance:

$$\sigma_M = \sum_{i=0}^{N} P(i)(i/N - \mu_M)^2 \tag{3.2}$$

  This describes the spread in the data. Images with urban structure usually has high standard variance (for the value of the gradient magnitude).

- Magnitude Skewness:

$$\text{Skewness} = \frac{1}{{\sigma_M}^3} \sum_{i=0}^{N} P(i)(i/N - \mu_M)^3 \tag{3.3}$$

  The skew measures the asymmetry (unbalance) about the mean in the gradient magnitude distribution. Region with urban structure should have high standard variance but low skew value.

- Magnitude Energy:

$$\text{Energy} = \sum_{i=0}^{N} [P(i)]^2 \tag{3.4}$$

  The energy measure tells us something about how gradient magnitude are distributed. This value gets smaller as the magnitude values are distributed across more gradient magnitude level values. A high energy means that the

number of gradient magnitude levels in the image is small. Urban region usually has higher energy value.

- Magnitude Entropy:

$$\text{Entropy} = -\sum_{i=0}^{N} P(i)\log_2(P(i)) \tag{3.5}$$

As the magnitude values are distributed among more magnitude levels, the entropy increases, while if the entropy value in a region is low we consider it as an urban area.

3.3 Gradient Orientation Features

The gradient orientation features described here are similar with the ones presented in [KH03]. For each pixel s, the gradients contained in a window W_s with size $n_s \times m_s$ centered around s form a histogram $\{H_s(i), i = 1, ..., N\}$ of 18 bins (with a bin width = 10 degree) for each site s, *i.e.*, instead of increasing the count +1 each time, we weight each count by the gradient magnitude at that location, as in [BP01]. Let E_δ be the magnitude of the histogram at the δ-th bin, and N be the total number of bins in the histogram. Then we smoothed the histogram using kernel smoothing: the smoothed histogram is given as

$$E'_\delta = \frac{\sum_{i=0}^{N} K((\delta - i)/h)E_i}{\sum_{i=0}^{N} K((\delta - i)/h)} \tag{3.6}$$

where K is a kernel function with bandwidth h. The kernel K is generally chosen to be a non-negative, symmetric function. From this smoothed histogram, we then compute the following features:

- Orientation Mean

$$\mu_R = \frac{1}{N}\sum_{\delta=0}^{N} E'_\delta \tag{3.7}$$

If the window W_s contains a smooth patch, the magnitude will be very small and the mean orientation of the histogram over all the bins will also be small. On the other hand, if W_s contains a few straight lines and/or edges embedded in a smooth background, a few bins will have significant peaks in the histogram in comparison to the other bins, and the mean value over all bins will be large.

- Central-Shift Moments

$$v_p = \frac{\sum_{i=0}^{N}(E'_\delta - v_0)^{p+1} H(E'_\delta - v_0)}{\sum_{i=0}^{N}(E'_\delta - v_0) H(E'_\delta - v_0)} \tag{3.8}$$

where $H(x)$ is the unit step function such that $H(x) = 1$ for $x > 0$, and 0 otherwise. The moment computation in Eq. (3.8) considers the contribution only from the bins having magnitude above the mean μ_R. Further, each bin value above the mean is linearly weighted by its distance from the mean so that the peaks far away from the mean contribute more. The moments v_p will be small if W_s contains smooth texture, otherwise it is large.

- Junction

$$\beta_c = |\sin(\delta_1 - \delta_2)| \tag{3.9}$$

Since man-made structures are usually of regular shape, the gradient orientation in such a region will show two or one clusters in histogram and the relation between the peaks of the histograms must thus contain useful information.

The peaks of the histogram are obtained simply by finding the local maximum of the smoothed histogram. If the texture in site s contains urban structures (like buildings, roads, etc.) the junction angle between two peaks should be around 90^0 (since the structure is likely to be perpendicular). Let δ_1 and δ_2 be the ordered orientations corresponding to the two highest estimated peaks such that $E'_{\delta_1} \geq E'_{\delta_2}$. We compute the junction measure between two peaks by the Eq. (3.9). A value close to 1 will represent an urban region.

3.4 Straight Line Features

Straight-line based features encode relatively strong structural information. Based on the results given by a LSD detector, we have first estimated, for each pixel, the local histogram of the line length and line contrast around a squared neighborhood window. We have obtained two histograms $\{H_s(i), i = 1, ..., N\}$ of 40 bins (with bin-width = 2 pixels) for line length and bin-width = 15 for line contrast). If a window W_s contains an urban patch, the mean of both line length and line contrast histograms, over all the bins, will be large, but due to human activity the entropy value will be small.

- Line Mean

$$\text{Length mean} = \sum_{i=0}^{N} P_{\text{length}}(i) \tag{3.10}$$

- Line Contrast

$$\text{Contrast mean} = \sum_{i=0}^{N} P_{\text{contrast}}(i) \tag{3.11}$$

- Line Length Entropy

$$\text{Length entropy} = -\sum_{i=0}^{N} P_{\text{length}}(i)\ \log_2(P_{\text{length}}(i)) \tag{3.12}$$

- Line Contrast Entropy

$$\text{Contrast entropy} = -\sum_{i=0}^{N} P_{\text{contrast}}(i)\ \log_2(P_{\text{contrast}}(i)) \tag{3.13}$$

3.5 GLCM Texture Features

The theoretical background of Grey-Level Co-occurrence Matrix (GLCM), has been already presented in Section 2.2.2. GLCM based textural features consider the relation between two neighboring pixels in one offset, as the second order statistics of a texture. The grey-value relationships in a target are transformed into the

co-occurrence matrix space by a given kernel mask (3×3, 5×5 or 7×7 depending on image resolution). In the transformation from the image space into the co-occurrence matrix space, the neighboring pixels in one or some of the eight defined directions can be used; normally, four directions such as 0^0, 45^0, 90^0, and 135^0 are initially regarded, and the reverse direction (negative direction) are included.

A number of texture features may be extracted from the GLCM (see [HSD73] and [CTH84]). We use the following notation: G is the number of grey levels used, μ is the mean value of P (the probability matrix), and μ_x, μ_y, α_x and α_y are the means and standard deviations of P_x and P_y. $P_x(i)$ is the i-th entry in the marginal-probability matrix obtained by summing the rows of $P(i,j)$:

$$P_x(i) = \sum_{j=0}^{G-1} P(i,j) \tag{3.14}$$

$$P_y(j) = \sum_{i=0}^{G-1} P(i,j) \tag{3.15}$$

$$\mu_x = \sum_{i=0}^{G-1} iP_x(i,j) \tag{3.16}$$

$$\mu_y = \sum_{j=0}^{G-1} jP_x(i,j) \tag{3.17}$$

$${\alpha_x}^2 = \sum_{i=0}^{G-1} (P_x(i) - \mu_x(i))^2 \tag{3.18}$$

$${\alpha_y}^2 = \sum_{j=0}^{G-1} (P_y(j) - \mu_y(j))^2 \tag{3.19}$$

$$P_{x+y}(k) = \underbrace{\sum_{i=0}^{G-1} \sum_{j=0}^{G-1}}_{i+j=k} P(i,j) \tag{3.20}$$

for $k = 0, 1, 2, \ldots, 2(G-1)$

$$P_{x-y}(k) = \underbrace{\sum_{i=0}^{G-1}\sum_{j=0}^{G-1} P(i,j)}_{|i-j|=k} \tag{3.21}$$

for $k = 0, 1, 2, \ldots, G-1$.

Therefore, the general GLCM texture measure is dependent upon kernel size and directionality. Known measures such as contrast, entropy, energy, dissimilarity, angular second moment (ASM) and homogeneity are expressed as follows:

- Homogeneity, Angular Second Moment (ASM) :

$$\text{ASM} = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1} P(i,j)^2 \tag{3.22}$$

  ASM is a measure of homogeneity of an image. A homogeneous scene (plane area) will contain only a few grey levels, giving a GLCM with only a few but relatively high values of $P(i,j)$. Thus, the sum of squares will be high.

- Contrast:

$$\text{Contrast} = \sum_{n=0}^{G-1} n^2 \Big\{ \underbrace{\sum_{i=0}^{G-1}\sum_{j=0}^{G-1} P(i,j)}_{|i-j|=n} \Big\}. \tag{3.23}$$

  Contrast is a measure of the local contrast of an image. The contrast is expected to be low if the grey levels in a region are similar. High contrast value indicates urban area.

- Entropy :

$$\text{Entropy} = -\sum_{i=0}^{G-1}\sum_{j=0}^{G-1} P(i,j)\ \log(P(i,j)) \tag{3.24}$$

  Entropy measures the randomness of a grey-level distribution. The entropy is expected to be high if the grey-levels are distributed randomly throughout the image. An urban region usually has low entropy.

- Correlation :

$$\text{Correlation} = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1} \frac{ij\,P(i,j) - \mu_x\,\mu_y}{\alpha_x\,\alpha_y} \tag{3.25}$$

Correlation provides a correlation between the two pixels in the pixel pair. The correlation is expected to be high if the grey levels of the pixel pairs are highly correlated. High correlation of grey-levels means this area contains man-made structures.

- Energy :

$$\text{Energy} = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1}(i-j)^2\,P(i,j) \tag{3.26}$$

Energy measures the number of repeated pairs. The energy is expected to be high if the occurrence of repeated pixel pairs is high, thus a plane area has a high energy value.

- Cluster Shade :

$$\text{Shade} = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1}(i+j-\mu_x-\mu_y)^3\,P(i,j) \tag{3.27}$$

- Cluster Prominence :

$$\text{Prom} = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1}(i+j-\mu_x-\mu_y)^4\,P(i,j) \tag{3.28}$$

CHAPTER 4

EXPERIMENTAL RESULTS

In this chapter, each individual (textural and structural) feature (presented in the previous chapter) is segmented, independently of the others, into two classes (*i.e.*, urban and non-urban areas) with a simple K-means clustering procedure on the four AVIRIS test images. This experimentation will allows us to select, in terms of visual evaluation, a subset of relevant features for our project. The most reliable features will be used in the final fusion procedure in order to provide a more reliable and accurate segmentation result.

In the following Figures, (a),(c),(e),(g), represent the original images compressed from hyperspectral data while (b),(d),(f),(h) are the results of a two-class clustering with the considered textural or structural feature.

4.1 GLCM Features

4.1.1 Cluster-Shade Average Feature

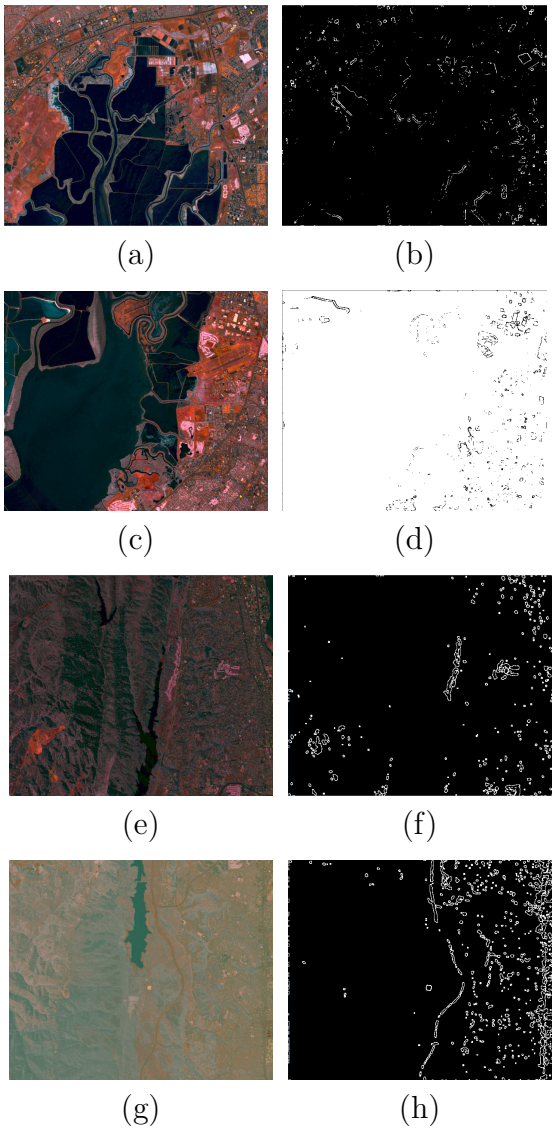


Figure 4.1: Cluster-shade average feature segmentations

4.1.2 Cluster-Shade Variance Feature

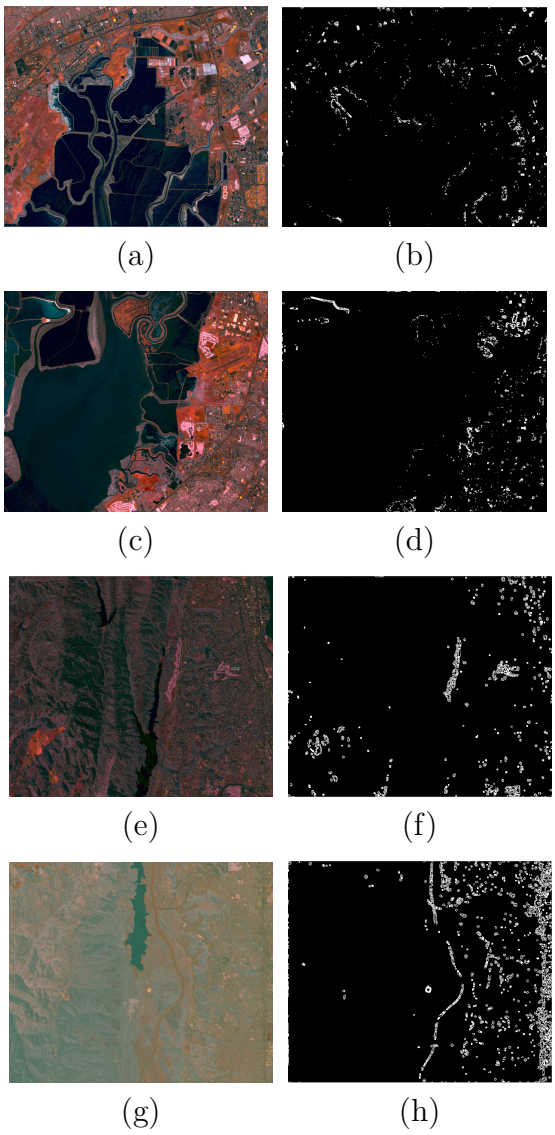


Figure 4.2: Cluster-shade variance feature segmentations

4.1.3 Cluster-Prominence Average Feature

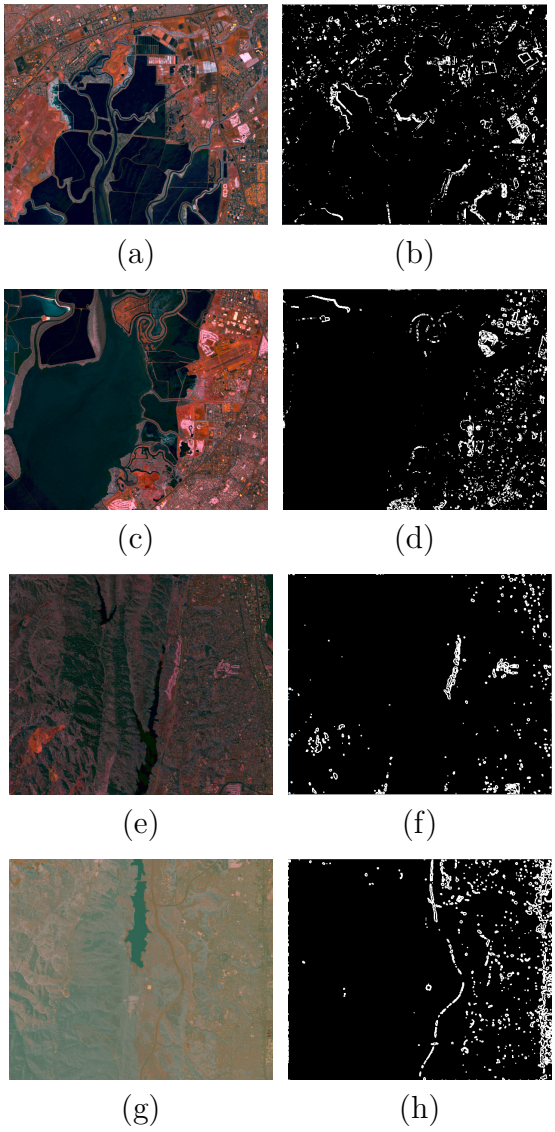


Figure 4.3: Cluster-prominence average feature segmentations

4.1.4 Cluster-Prominence Variance Feature

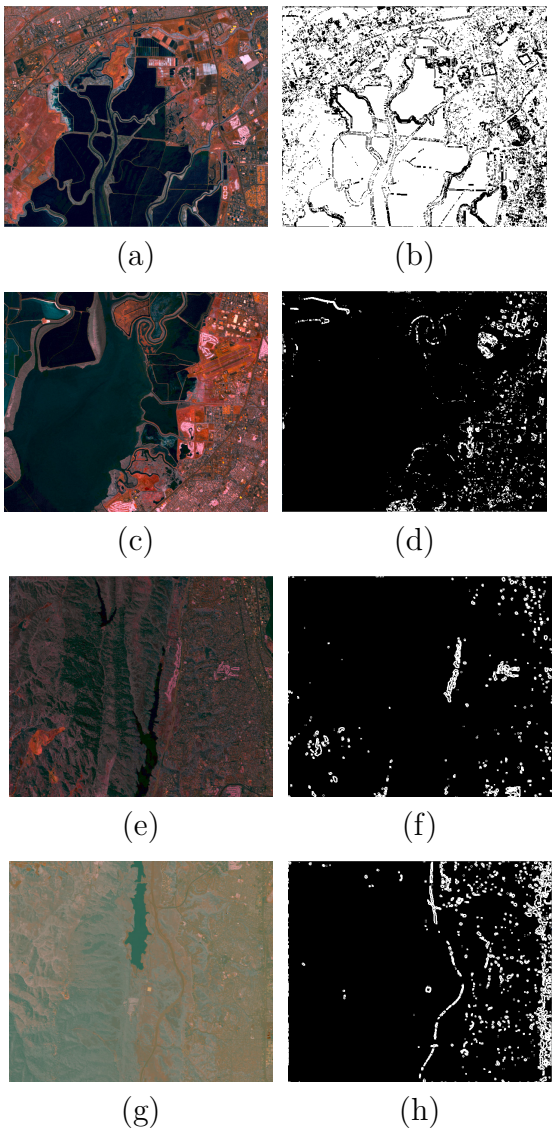


Figure 4.4: Cluster-prominence variance feature segmentations

4.1.5 Contrast Average Feature

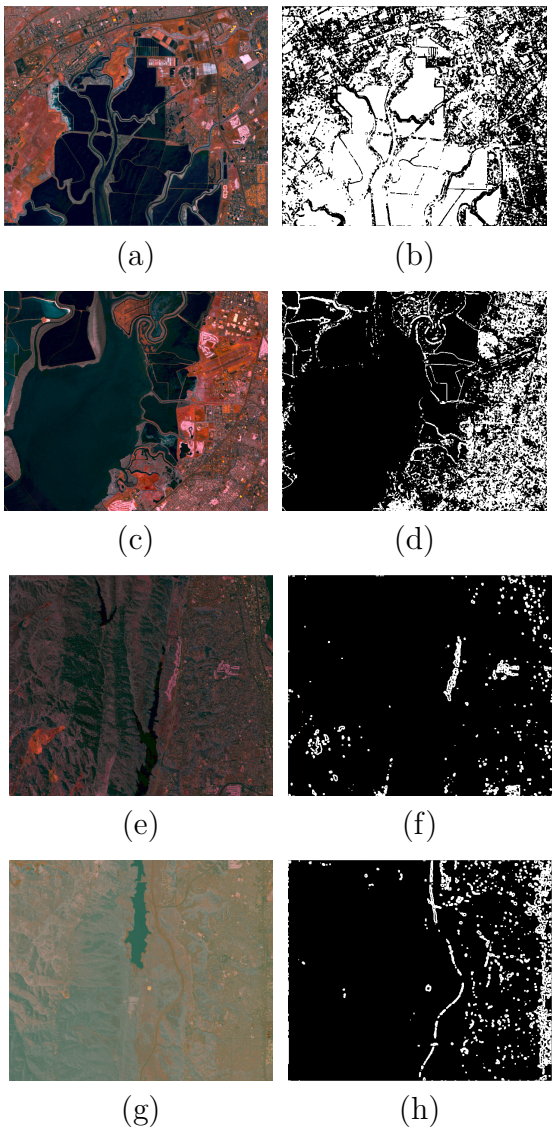


Figure 4.5: Contrast average feature segmentations

4.1.6 Contrast-Variance Feature

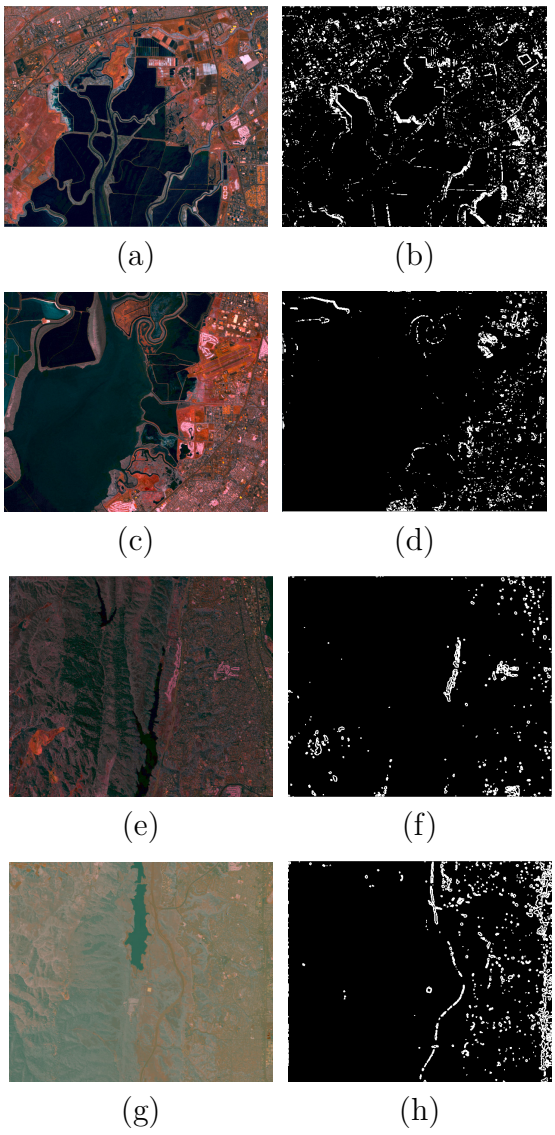


Figure 4.6: Contrast-variance feature segmentations

4.1.7 Energy-Average Feature

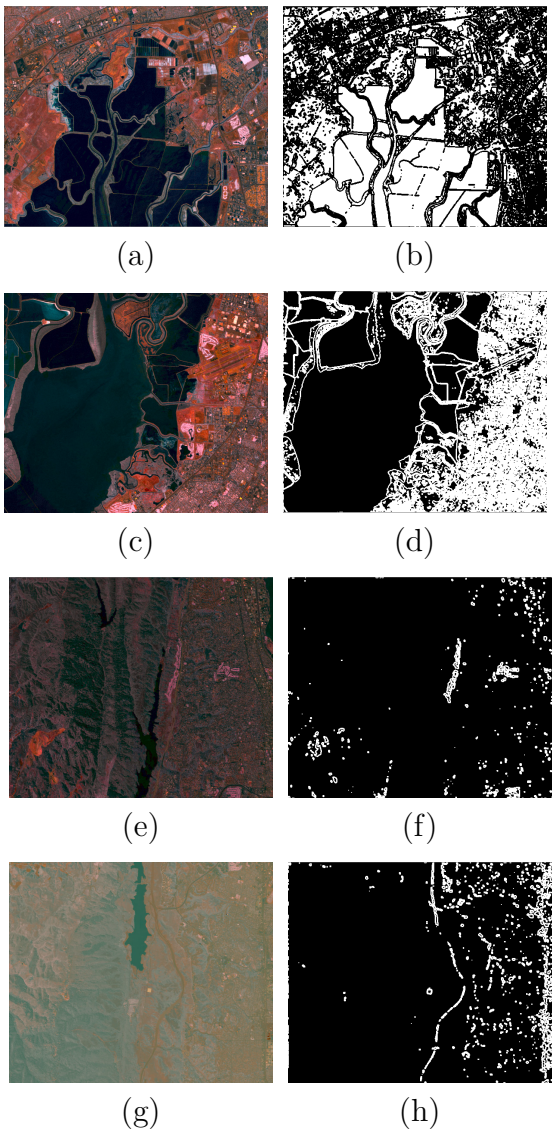


Figure 4.7: Energy-average feature segmentations

4.1.8 Energy-Variance Feature

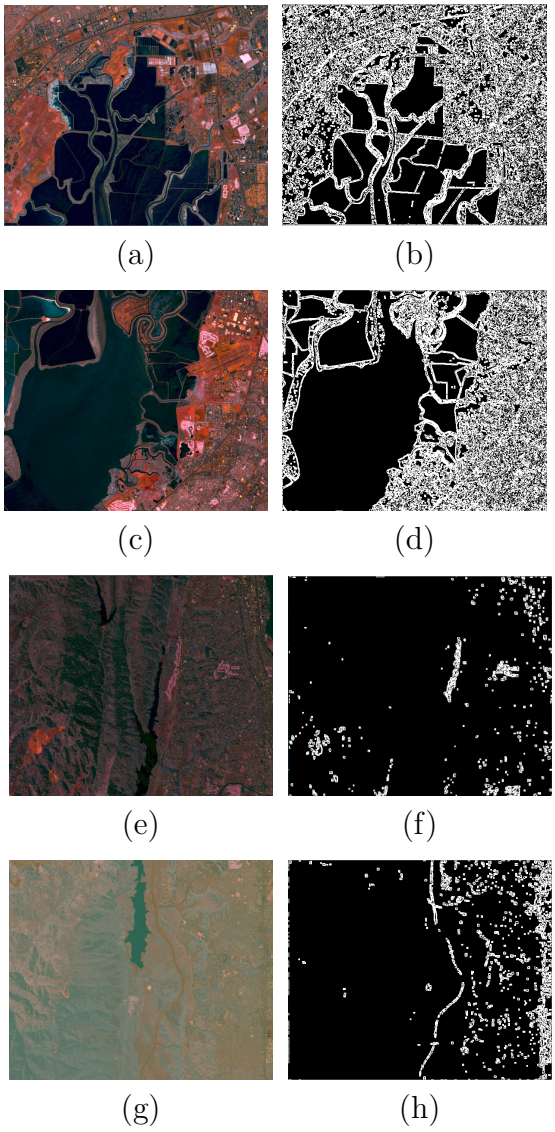


Figure 4.8: Energy-variance feature segmentations

4.1.9 Entropy-Average Feature

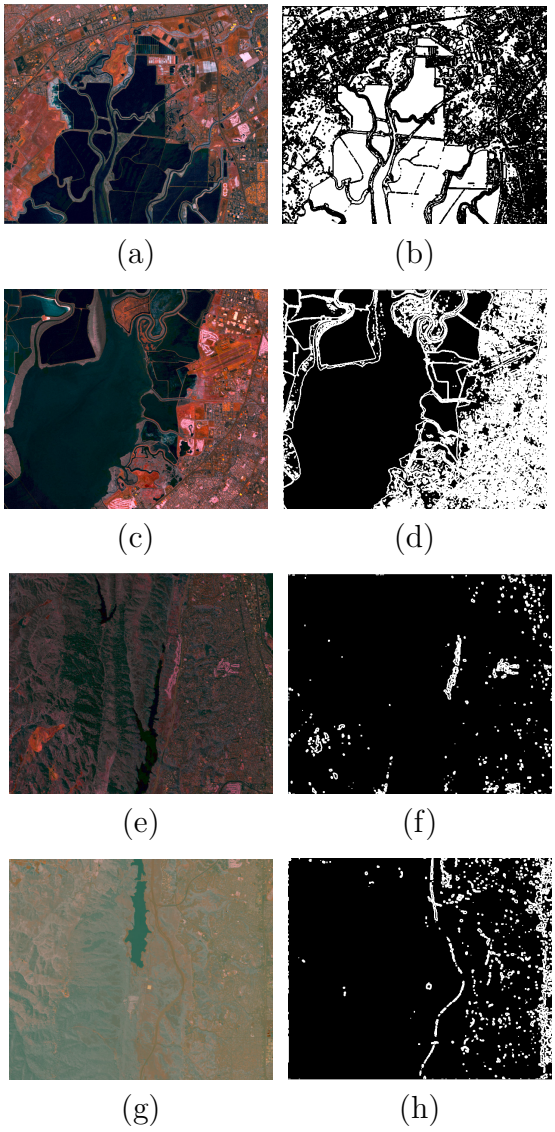


Figure 4.9: Entropy-average feature segmentations

4.1.10 Entropy-Variance Feature

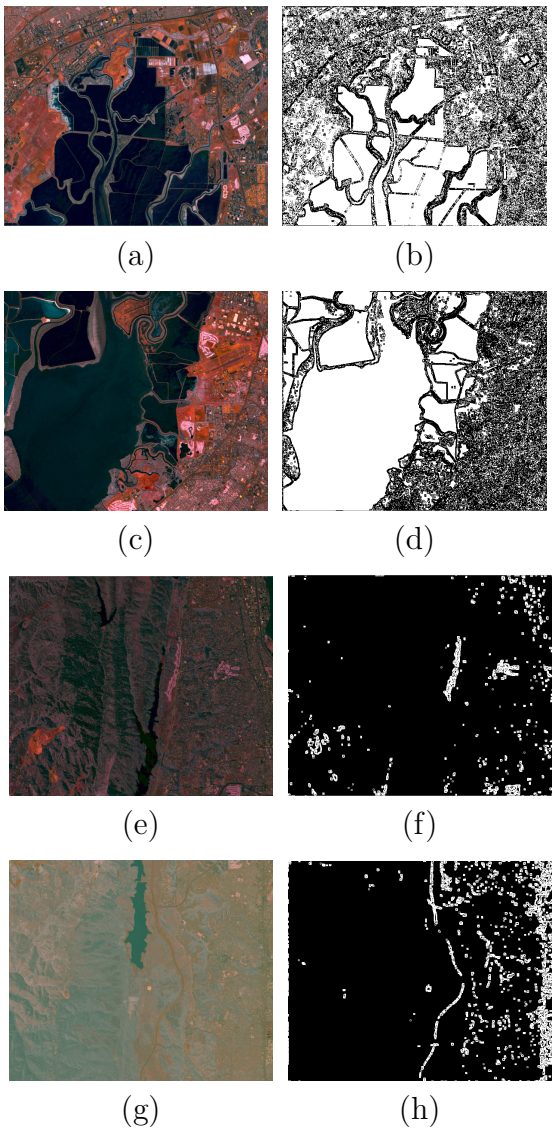


Figure 4.10: Entropy-variance feature segmentations

4.1.11 Homogeneity-Average Feature

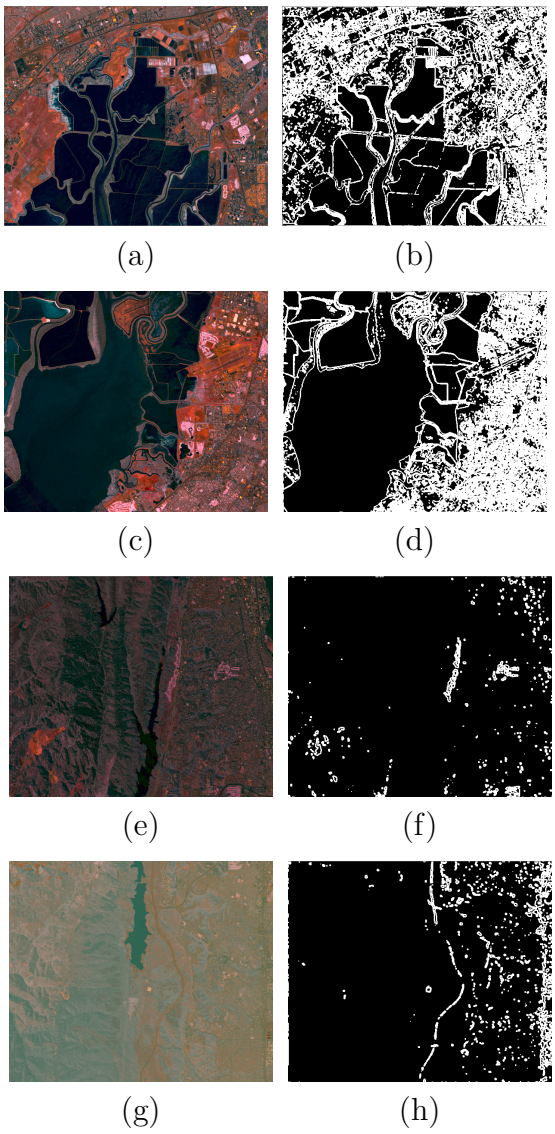


Figure 4.11: Homogeneity-average feature segmentations

4.1.12 Homogeneity-Variance Feature

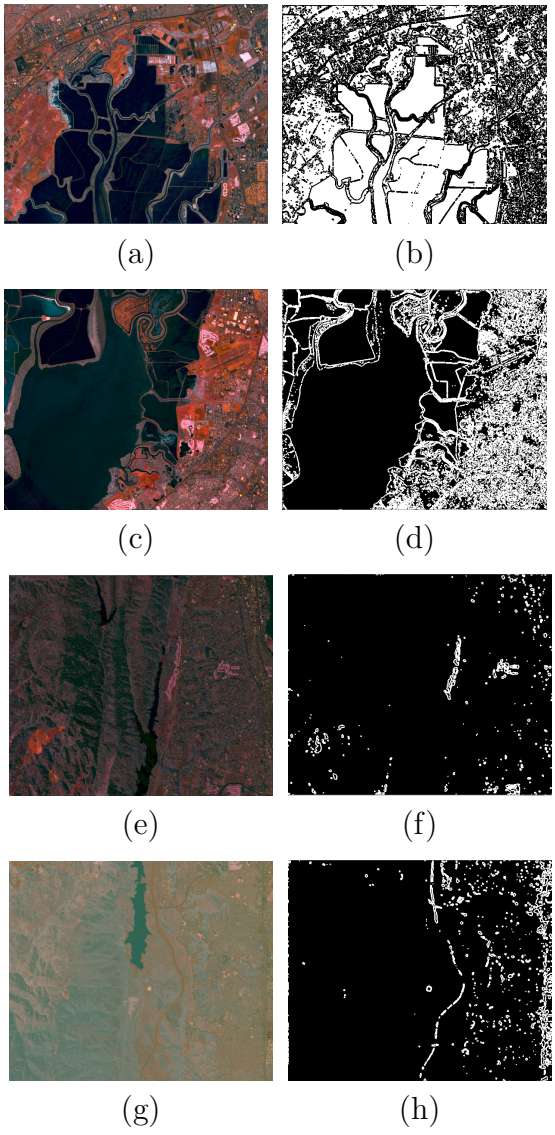


Figure 4.12: Homogeneity-variance feature segmentations

4.2 Gradient Magnitude Features

4.2.1 Energy Feature

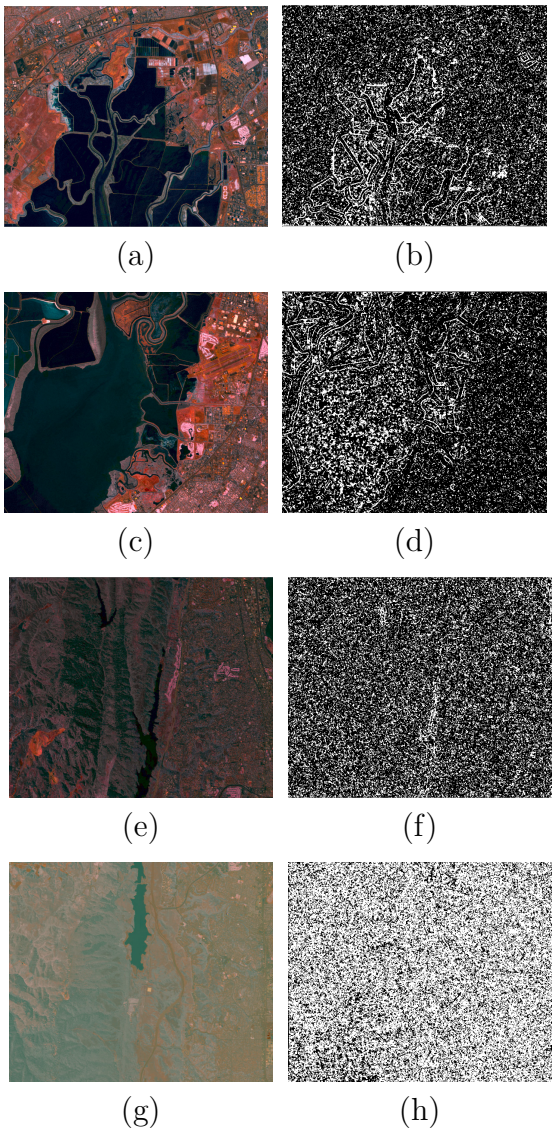


Figure 4.13: Energy feature segmentations

4.2.2 Entropy Feature

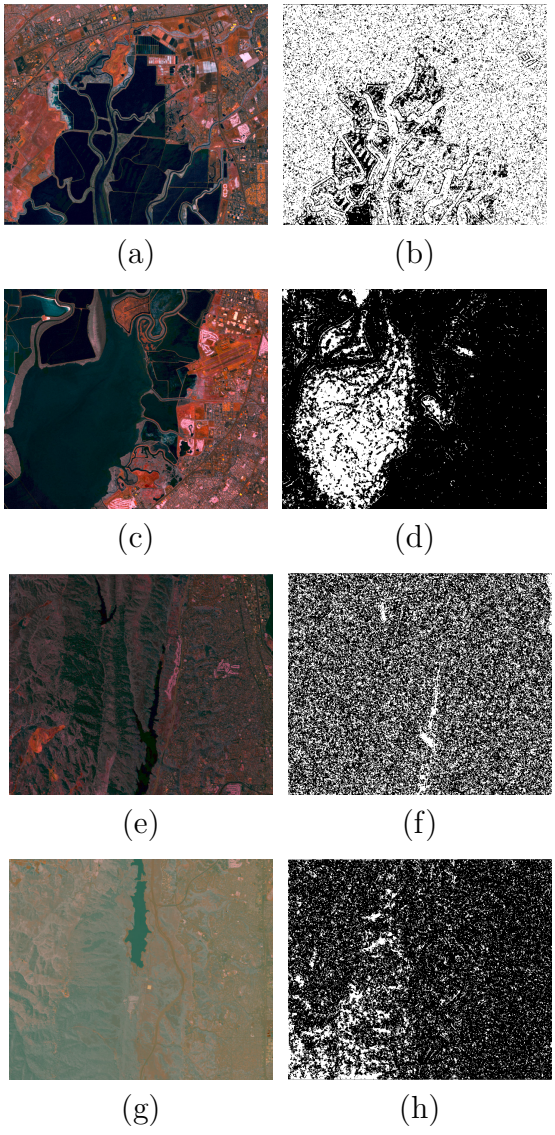
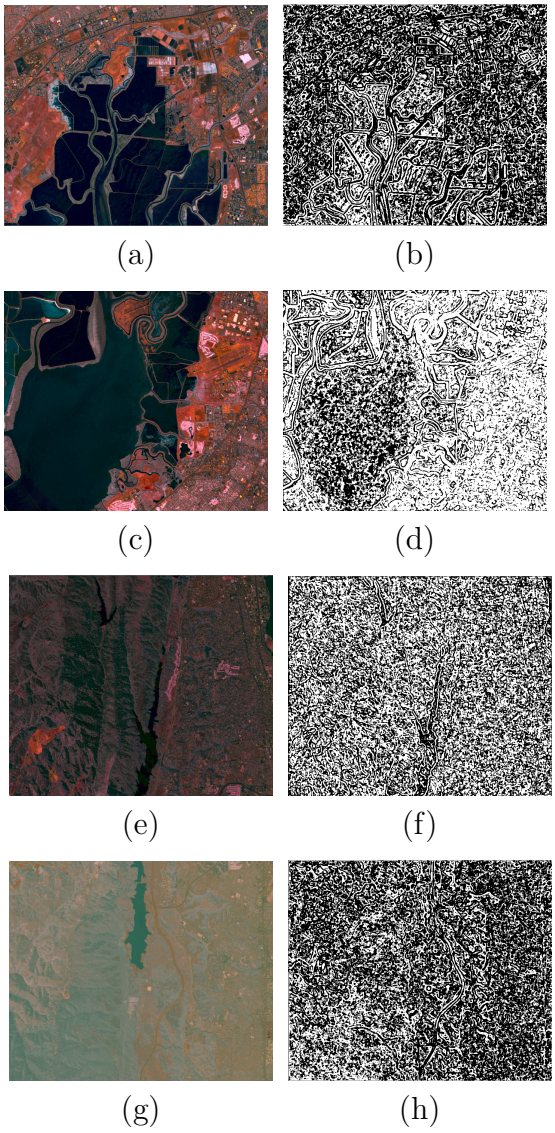


Figure 4.14: Entropy feature segmentations

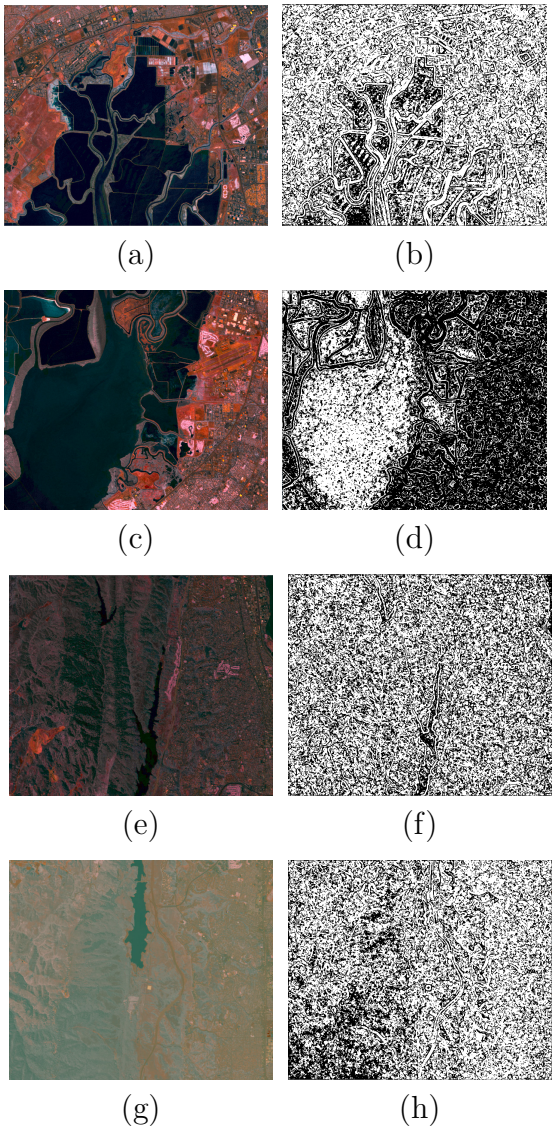
4.2.3 Mean Feature



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 4.15: Mean feature segmentations

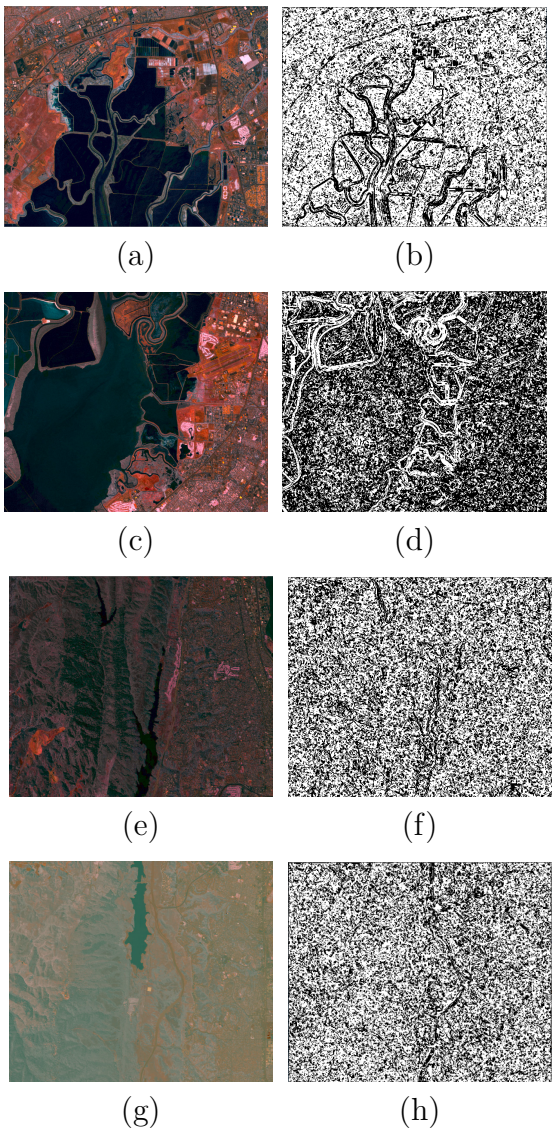
4.2.4 Skewness Feature



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 4.16: Skewness feature segmentations

4.2.5 Smooth Feature



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 4.17: Smooth feature segmentations

4.3 Gradient Orientation Features

4.3.1 Mean Orientation Feature

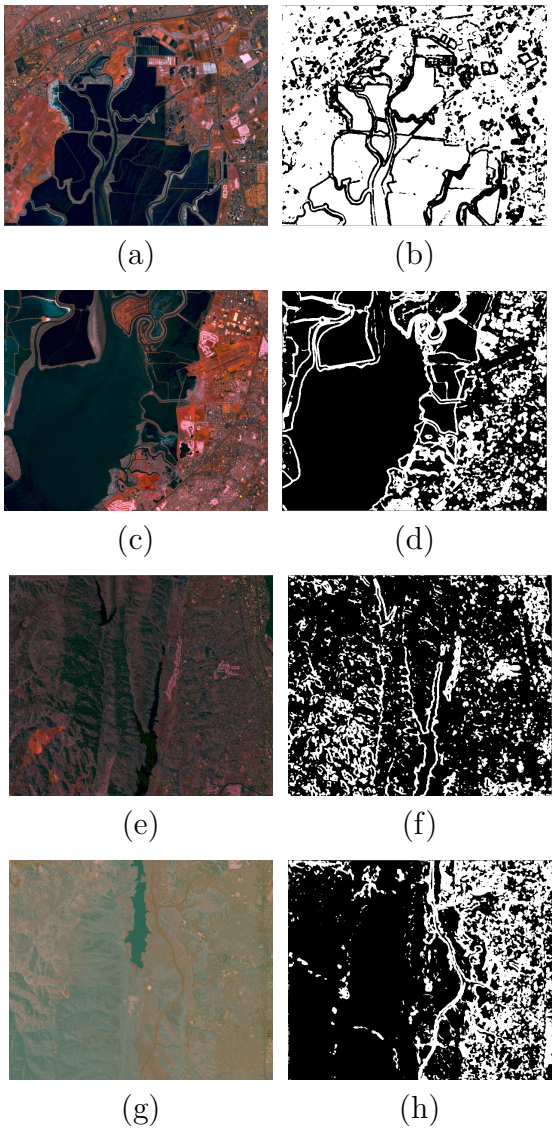


Figure 4.18: Mean orientation feature segmentations

4.3.2 Second Moment Feature

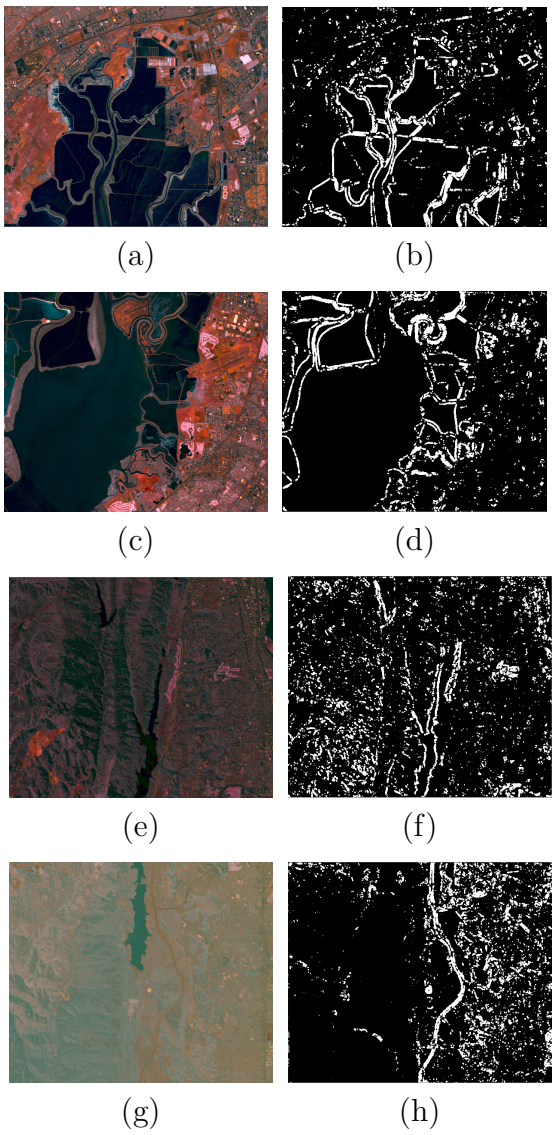
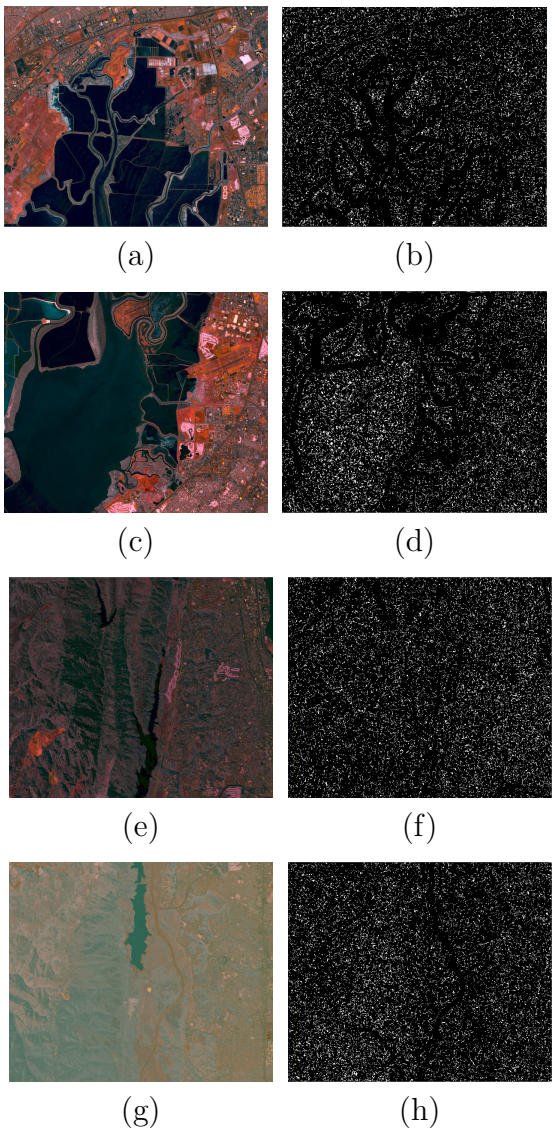


Figure 4.19: Second moment feature segmentations

4.3.3 Junction Feature

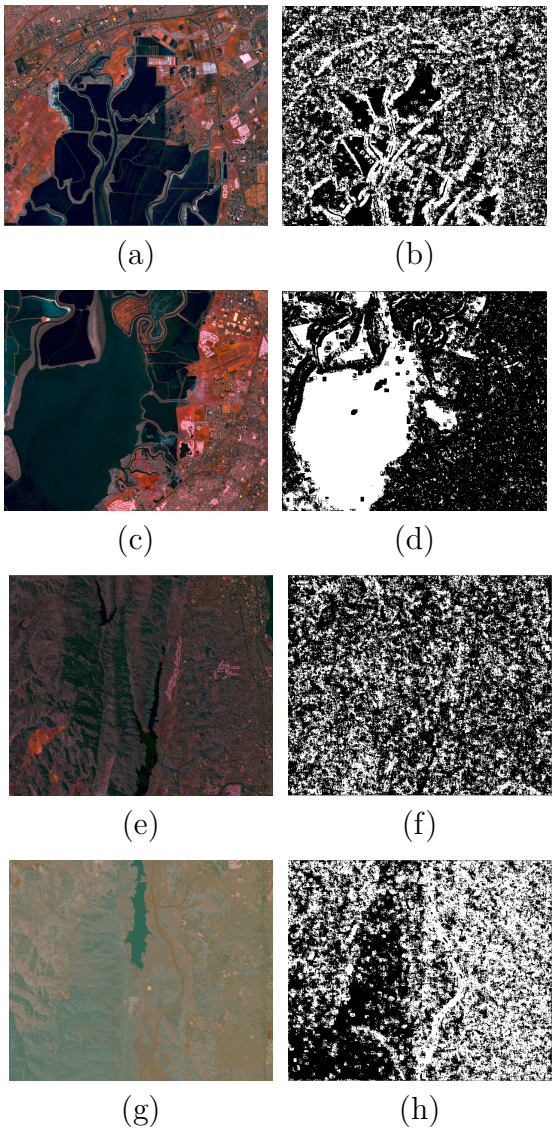


(a) (b) (c) (d) (e) (f) (g) (h)

Figure 4.20: Junction feature segmentations

4.4 Straight Line Features

4.4.1 Length Feature



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 4.21: Length feature segmentations

4.4.2 Contrast Feature

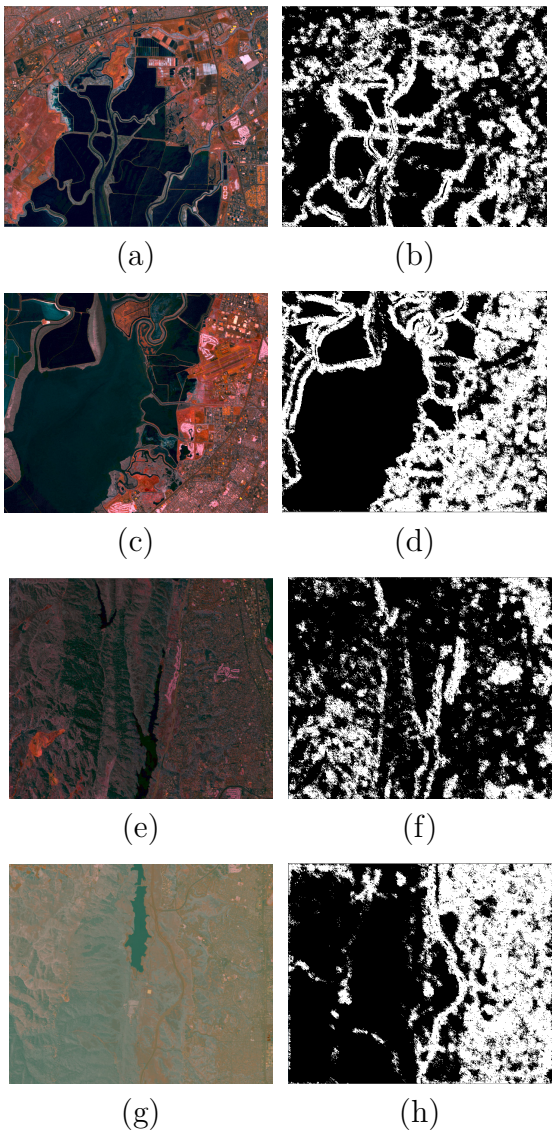


Figure 4.22: Contrast feature segmentations

4.4.3 Length Entropy Feature

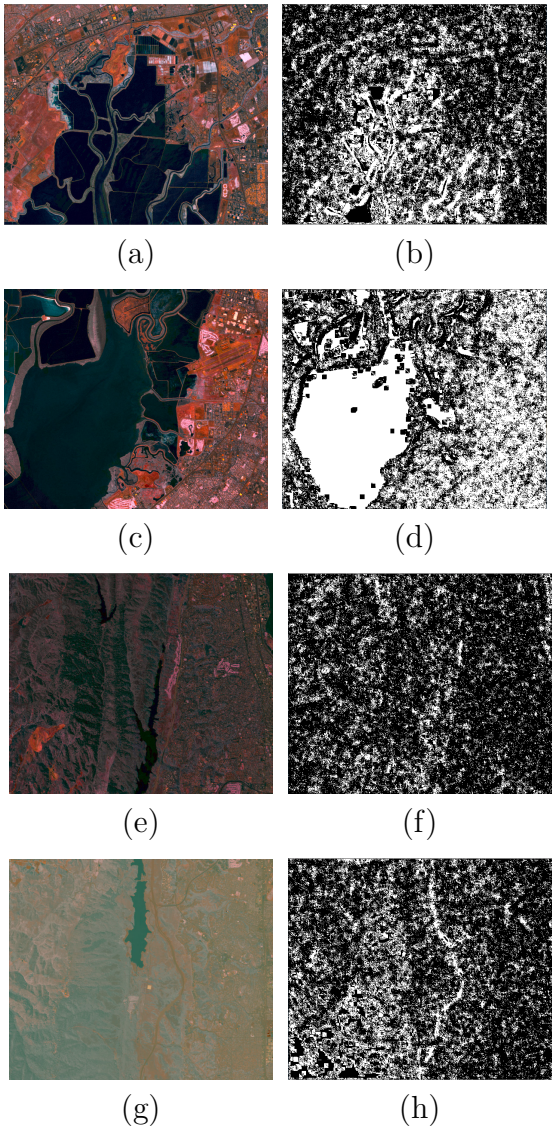


Figure 4.23: Length entropy feature segmentations

4.4.4 Contrast Entropy Feature

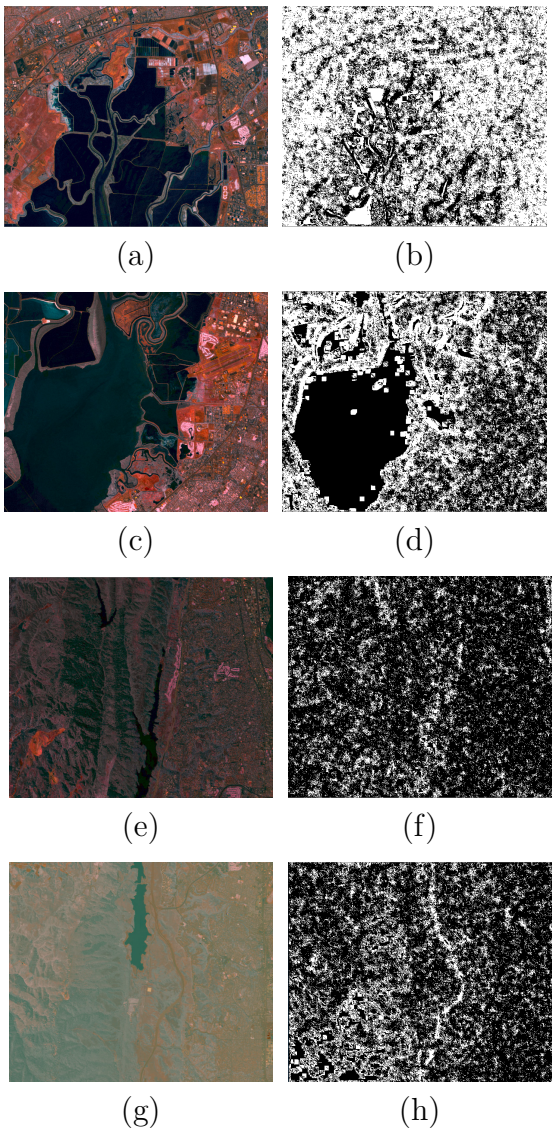


Figure 4.24: Contrast entropy feature segmentations

4.5 Feature Selection

We have tested four satellite images showing a wide variety of homogeneous regions such as lake, plane area, man-made structure (road, building, etc), mountain, forest, etc.. According to the segmentation results, in terms of visual evaluation, we decided to select 14 features from among these 24 features, namely,

- The GLCM cluster shade average
- The GLCM cluster shade variance
- The GLCM cluster prominence average
- The GLCM cluster prominence variance
- The GLCM contrast average
- The GLCM contrast variance
- The GLCM energy average
- The GLCM entropy average
- The GLCM entropy variance
- The GLCM homogeneity average
- The GLCM homogeneity variance
- The gradient orientation average
- The gradient orientation second moment
- The straight line contrast

4.6 Fusion Result

We present now the fusion results obtained by the model proposed in [Mig08] [Mig10a] on the four AVIRIS test images. In the following four result pages, (a) is the original compressed image, while (b)-(o) designate the two-class segmentation map for each of the above-mentioned feature. Finally (p),(q),(r) give the fused segmentation map result obtained with the Class Label Histogram (CLH) based fusion model [Mig08], the Probabilistic Rand Index (PRI) based fusion model [Mig10a] and finally the High Dimensional Vector (HDV) based fusion model on respectively MoffetField2, MoffetField3, JasperRidge1 and JasperRidge2 images.

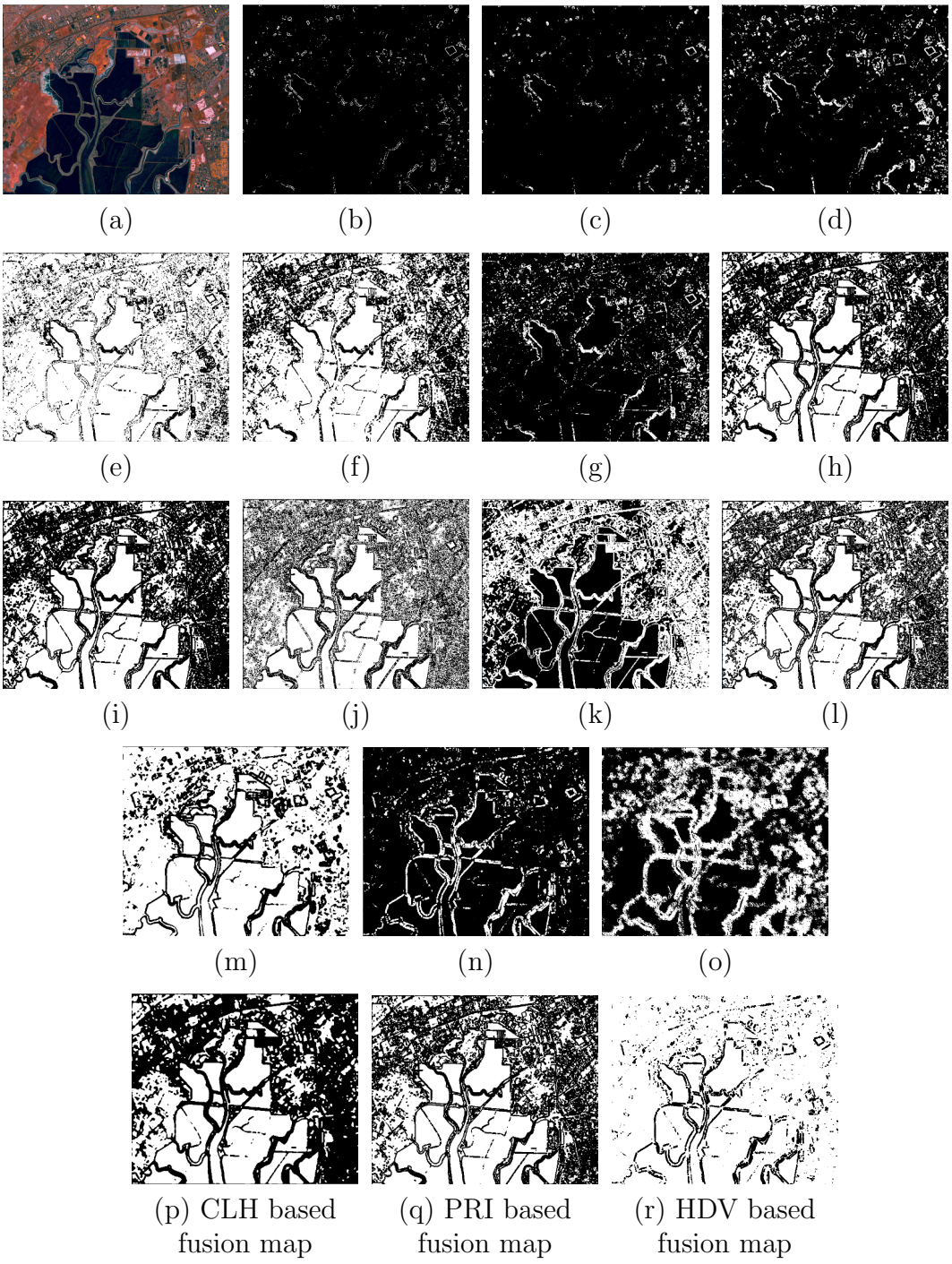
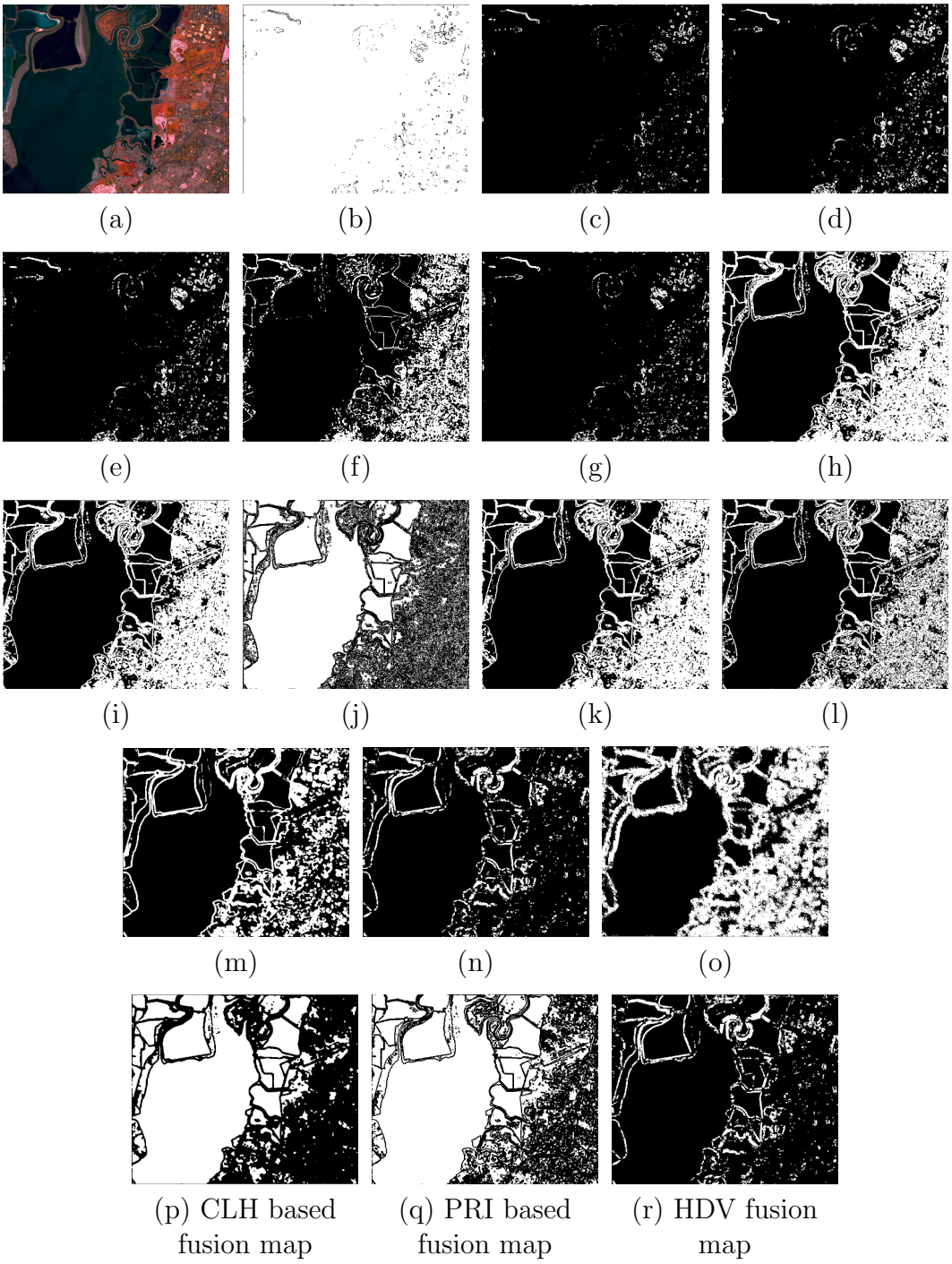


Figure 4.25: MoffetField2 fusion procedure



(a) (b) (c) (d)

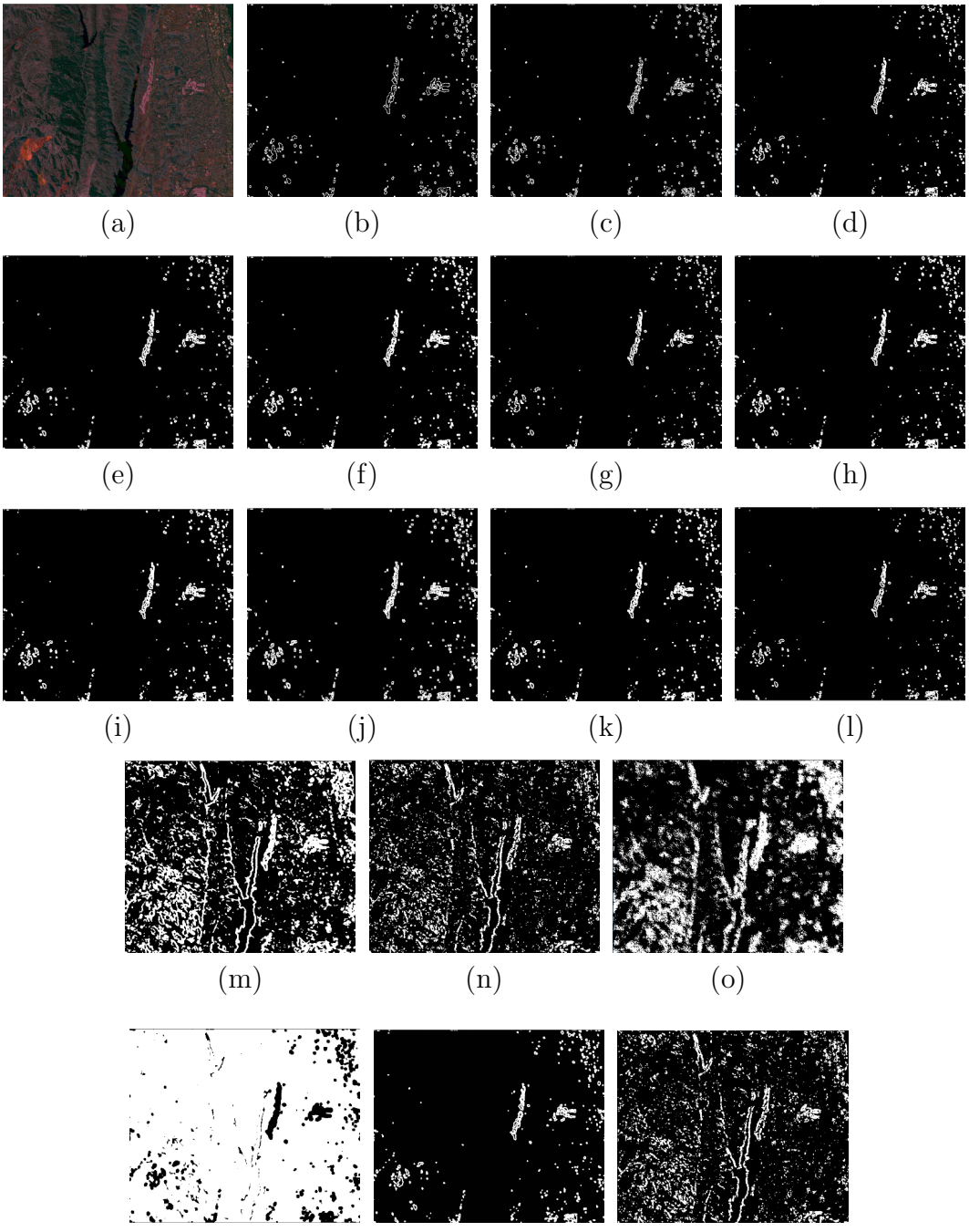
(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o)

(p) CLH based fusion map (q) PRI based fusion map (r) HDV fusion map

Figure 4.26: MoffetField3 fusion procedure



(a) (b) (c) (d)

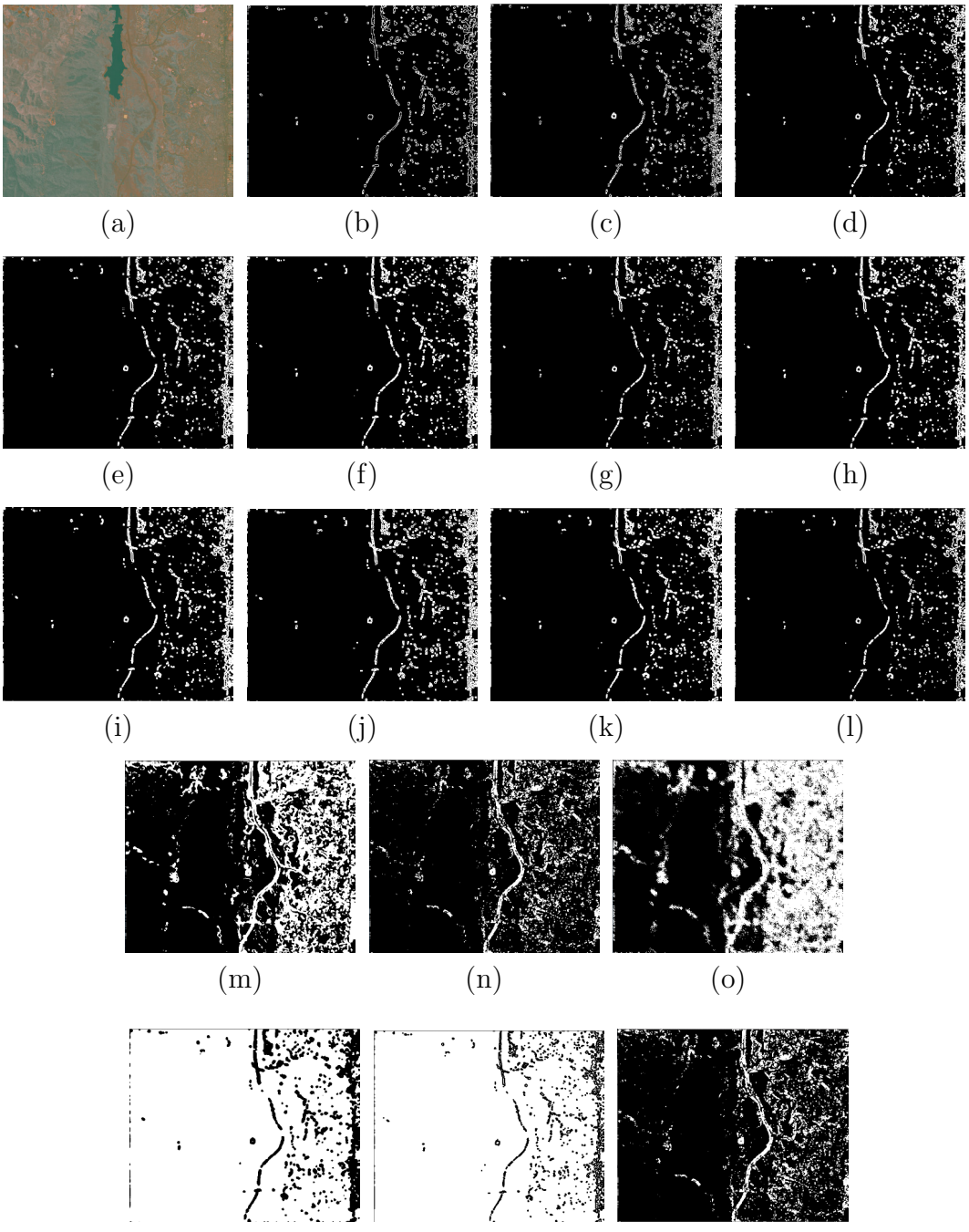
(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o)

(p) CLH based fusion map (q) PRI based fusion map (r) HDV based fusion map

Figure 4.27: JasperRidge1 fusion procedure



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o)

(p) CLH based fusion map (q) PRI based fusion map (r) HDV based fusion map

Figure 4.28: JasperRidge2 fusion procedure

4.7 Conclusion

It can be seen that the images are correctly segmented into two classes (more details are given in the following chapter and in Annexes) by the two fusion procedures. The urban areas are not mistaken for mountainous areas or forests. Scattered (JasperRidge) or dense urban areas (Moffet Field) are also clearly distinguished. More precisely, the PRI based fusion model gives a better and more accurate result. The fusion method also spent a little more time than the CLH based fusion technique. We have also found that the HDV fusion model didn't provide us with a satisfactory result, (almost certainly due to the curse of dimensionality, the rescaling problem and the reasons already explained in Section 2.3).

CHAPTER 5

COMPARISON AND DISCUSSION

5.1 Experimental Environment

We have implemented this project in C/C++ code with OpenCV (Mac OS based on 2.4 GHZ Intel core 2 duo processor and 4 GB DDR3 memory). Source data are from http://www.iro.umontreal.ca/ mignotte/ in the format .rfl. Dimension of data is $512 \times 614 \times 224$. The computational time, in seconds, is indicated for each experiment.

5.2 Results Comparison

In this section we have first summarized the main results previously obtained *i.e.*,

1. Using the Label Class Histogram Based Fusion Model
2. Using the PRI Based Fusion Model
3. Using simply a high dimensional feature vector combining all the features

with respectively two and three classes for the final segmentation map, and we conclude this chapter with a brief discussion.

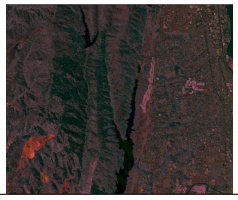
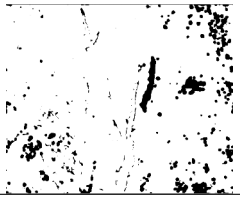
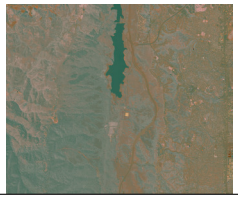
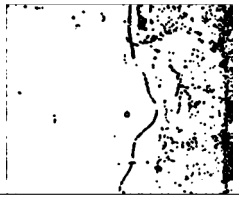
Image	Av. Run. Time	Compressed Image	Segmentation Image
MoffetField2.rfl	≈ 687 sec.		
MoffetField3.rfl	≈ 659 sec.		
JasperRidge1.rfl	≈ 738 sec.		
JasperRidge2.rfl	≈ 711 sec.		

Table 5.1: CLH based fusion procedure (2 classes)

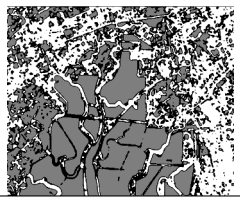
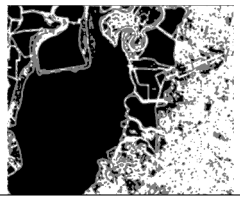
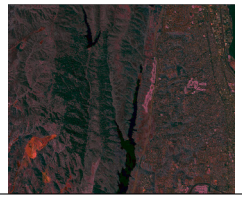
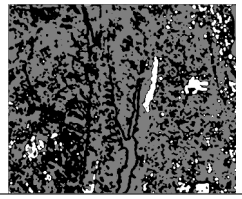
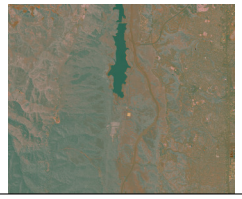
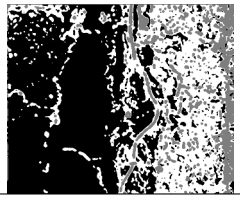
Image	Av. Run. Time	Compressed Image	Segmentation Image
MoffetField2.rfl	≈ 714 sec.		
MoffetField3.rfl	≈ 704 sec.		
JasperRidge1.rfl	≈ 710 sec.		
JasperRidge2.rfl	≈ 650 sec.		

Table 5.2: CLH based fusion procedure (3 classes)

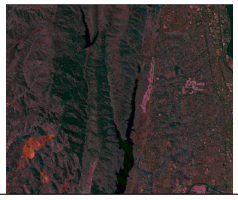
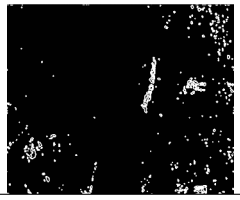
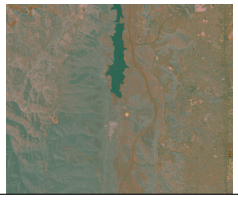
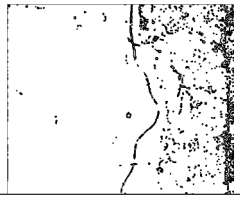
Image	Av. Run. Time	Compressed Image	Segmentation Image
MoffetField2.rfl	≈ 746 sec.		
MoffetField3.rfl	≈ 756 sec.		
JasperRidge1.rfl	≈ 790 sec.		
JasperRidge2.rfl	≈ 776 sec.		

Table 5.3: PRI based fusion procedure (2 classes)

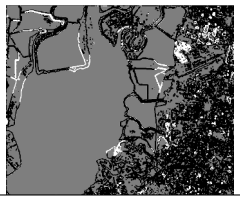
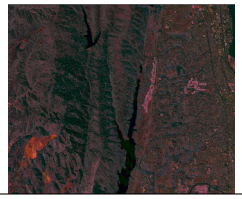
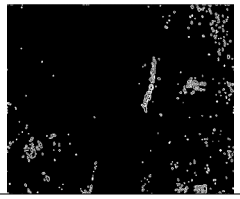
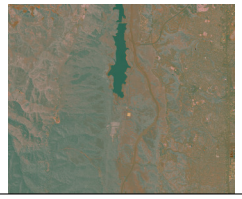
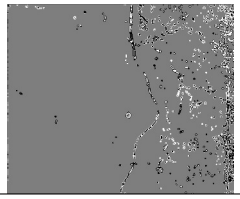
Image Data	Av. Run. Time	Compressed Image	Segmentation Image
MoffetField2.rfl	≈ 848 sec.		
MoffetField3.rfl	≈ 752 sec.		
JasperRidge1.rfl	≈ 847 sec.		
JasperRidge2.rfl	≈ 809 sec.		

Table 5.4: PRI based fusion procedure (3 classes)

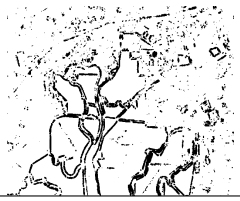
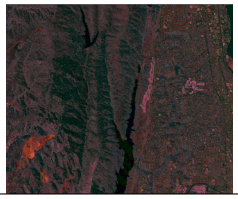
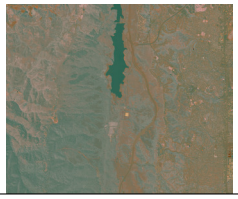
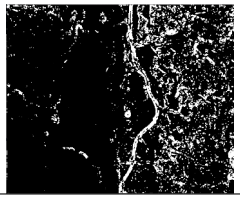
Image Data	Av. Run. Time	Compressed Image	Segmentation Image
MoffetField2.rfl	≈ 666 sec.		
MoffetField3.rfl	≈ 654 sec.		
JasperRidge1.rfl	≈ 699 sec.		
JasperRidge2.rfl	≈ 628 sec.		

Table 5.5: Segmentation by high dimensional features vector (2 classes)

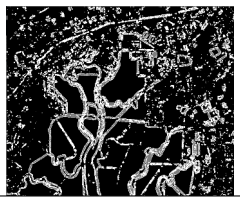
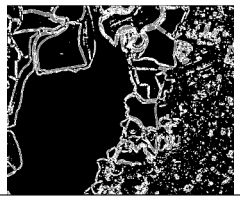
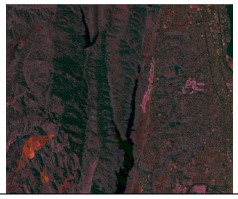
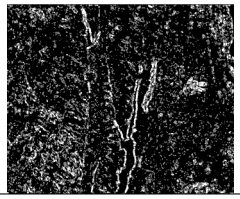
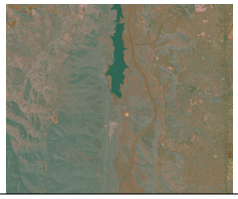
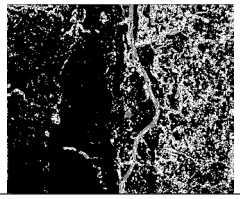
Image Data	Av. Run. Time	Compressed Image	Segmentation Image
MoffetField2.rfl	≈ 663 sec.		
MoffetField3.rfl	≈ 654 sec.		
JasperRidge1.rfl	≈ 699 sec.		
JasperRidge2.rfl	≈ 638 sec.		

Table 5.6: Segmentation by high dimensional features vector (3 classes)

It can be seen that the precision of final result also depends on the number of classes set *a priori* before the fusion procedure. In our application, an interesting approach consists in segmenting the image into three classes instead of two classes. In this case, the three segmented regions are URBAN REGION, NON-URBAN REGION and WEAK STRUCTURE OR NON-DENSE URBAN AREA.

CHAPTER 6

CONCLUSION

In this master's thesis, we have compared three different approaches for segmenting hyperspectral images into two classes (namely, urban and non-urban areas) from a set of structural and textural features and a final clustering procedure. We have compared the classical approach which consists in gathering all the features in a high dimensional feature vector along with a final clustering procedure and two different fusion models. These two latter approaches first rely on the previous classification of each individual feature by a simple K-means clustering procedure and aim at finally combining these segmentation maps in a final (more reliable and accurate) segmentation result. These last two approaches give the most promising results. This is certainly due to the curse of dimensionality, the rescaling problem and the reasons already explained in Section 2.3.

The final classification procedure allows us to efficiently detect the URBAN AREAS and the RURAL AREAS (with various backgrounds) when we consider two classes and to discriminate between the class DENSE NON-URBAN REGION and NON DENSE URBAN AREA when we consider three classes.

There are still many issues that could improve this unsupervised classification method. The first and foremost would consist in taking into account, in addition to the textural and structural features used in this project, some spectral features. Indeed, since the dense urban regions are mainly composed of a few materials such as concrete, steel and wood (for the buildings and houses), coal tar (for the roads) and aluminium (for the car and light metallic structures), it would be very interesting to use this *a priori* knowledge in order to identify spatial regions whose spectrum, in the raw uncompressed hyperspectral images, is really a mixture of these five components.

BIBLIOGRAPHY

[BAF05] C. M. Bachmann, T.L. Ainsworth, and R.A. Fusina. Exploiting manifold geometry in hyperspectral imagery. *IEEE Trans. Geosci. Remote Sensing*, 43(3):441–454, March 2005.

[BHR86] J. B. Burns, A. R. Hanson, and E. M. Riseman. Extracting straight lines. *IEEE Trans. Pattern Anal. Machine Intell.*, 8(4):425–455, 1986.

[BP01] W. A. Barrett and K. D. Petersen. Houghing the hough: Peak collection for detection of corners, junctions and line intersections. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR'01)*, volume 2, pages 302–309, 2001.

[CD99] C.-I. Chang and Q. Du. Interference and noise adjusted principal components analysis. *IEEE Trans. Geosci. Remote Sensing*, 37(9):2387–2396, September 1999.

[CRHW09] M. Cui, A. Razddan, J. Hu, and P. Wonka. Interactive hyperspectral image visualization using convex optimization. *IEEE Trans. Geosci. Remote Sensing*, 47(6):1673–1684, June 2009.

[CTH84] R.W. Conners, M.M. Trivedi, and C.A. Harlow. Segmentation of a high-resolution urban scene using texture operators. *Computer Vision, Graphics, and Image Processing*, 25:273–310, 1984.

[DMM00] A. Desolneux, L. Moisan, and J.-M. Morel. Meaningful alignments. *International Journal of Computer Vision*, 40(1):7–23, 2000.

[DQWR03] H. Du., H. Qi, X. Wang, and R. Ramanath. Band selection using independent component analysis for hyperspectral image processing. In *Proc. 32nd Applied Imagery Pattern Recognition Workshop*, pages 93–99, Washington, USA, October 2003.

[DRCM08] Q. Du, N. Raksuntorn, S. Cai, and R.J. Moorhead. Color display for hyperspectral imagery. *IEEE Trans. Geosci. Remote Sensing*, 46(6):1858–1866, June 2008.

[DT05] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR'05)*, June 2005.

[FL95] C. Faloutsos and K. Lin. Fastmap: a fast algorithm for indexing, data-mining and visualization. In *In Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 163–174, May 1995. source code of FastMap available on his webpage.

[HSD73] R.M. Haralick, K. Shanmugam, and ItsHak Dinstein. Textural features for image classification. *IEEE Trans. Syst., Man, Cybern.*, 6, November 1973.

[Hyp] www.hyvista.com.

[JG05] N. P. Jacobson and M. R. Gupta. Design goals and solutions for display of hyperspectral images. *IEEE Trans. Geosci. Remote Sensing*, 43(11):2684–2692, November 2005.

[JGC07] N. P. Jacobson, M. R. Gupta, and J.B. Cole. Linear fusion of image sets for display. *IEEE Trans. Geosci. Remote Sensing*, 45(10):3277–3288, October 2007.

[JR99] X. Jia and J. A. Richard. Segmented principal components transformation for efficient hyperspectral remote-sensing image display and classification. *IEEE Trans. Geosci. Remote Sensing*, 37(1):538–542, January 1999.

[JRACVR07] L. O. Jimenez-Rodriguez, E. Arzuaga-Cruz, and M. Velez-Reyes. Unsupervised linear feature-extraction methods and their effects in

the classification of high-dimensional data. *IEEE Trans. Geosci. Remote Sensing*, 45(2):469 – 483, January 2007.

[KC10] K. Kotwal and S. Chaudhuri. Visualization of hyperspectral images using bilateral filtering. *IEEE Trans. Geosci. Remote Sensing*, 48(5):2308 – 2316, April 2010.

[KH03] S. Kumar and M. Hebert. Man-made structure detection in natural im ages using a causal multiscale random field. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR'03)*, volume 1, pages 119–126., 2003.

[KIR00] V. Karathanassi, C. Iossidis, and D. Rokos. A texture-based classification method for classifying built areas according to their density. *International Journal of Remote Sensing*, 21(9):1807–1823, 2000.

[LN04] J. Li and R. M. Narayanan. Integrated spectral and spatial information mining in remote sensing imagery. *IEEE Trans. Geosci. Remote Sensing*, 42(3):673–685, March 2004.

[Mig08] M. Mignotte. Segmentation by fusion of histogram-based K-means clusters in different color spaces. *IEEE Trans. Image Processing*, 17(5):780–787, May 2008.

[Mig10a] M. Mignotte. A label field fusion bayesian model and its penalized maximum rand estimator for image segmentation. *IEEE Trans. Image Processing*, 19(6):1610–1624, June 2010.

[Mig10b] M. Mignotte. A multiresolution markovian fusion model for the color visualization of hyperspectral images. *IEEE Trans. Geosci. Remote Sensing*, 48(12):4236–4247, December 2010.

[Mig12] M. Mignotte. A bi-criteria optimization approach based dimensionality reduction model for the color display of hyperspectral images. *IEEE Trans. Geosci. Remote Sensing*, 50(2):501–513, January 2012.

[MLM95] S.K. Mitra, H. Li, and B.S. Manjunath. Multisensor image fusion using the wavelet transform. *Comp. Vis., Graph., Image Process.*, 57(3):627–640, 1995.

[MRR+53] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21:1087–1092, 1953.

[MSB07] A. Mohan, G. Sapiro, and E. Bosh. Spatially coherent nonlinear dimensionality reduction and segmentation of hyperspectral images. *IEEE Geosci. Remote Sensing Letters*, 4(2):206–210, April 2007.

[Nas] http://aviris.jpl.nasa.gov.

[Nat] www.nature.com.

[ND05] J.M.P. Nascimento and J.M.B. Dias. Does independent component analysis play a role in unmixing hyperspectral data ? *IEEE Trans. Geosci. Remote Sensing*, 43(1):175–187, January 2005.

[Pla05] J. C. Platt. Fastmap, metricmap, and landmark MDS are all nystrom algorithms. In *In Proceedings of 10th International Workshop on Artificial Intelligence and Statistics (AISTATS)*, pages 261–268, 2005.

[RK82] A. Rosenfeld and A. C. Kak. *Digital Picture Processing*. New York: Academic Press, 1982.

[Rog94] R.E. Roger. A faster way to compute the noise-adjusted principal component transform matrix. *IEEE Trans. Geosci. Remote Sensing*, 32(6):1194–1196, June 1994.

[RWK97] S. K. Rogers, T. A. Wilson, and M. Kabrisky. Perceptual-based image fusion for hyperspectral data. *IEEE Trans. Geosci. Remote Sensing*, 35(4):1007–1017, July 1997.

[SG84] P. Switzer and A. A. Green. Min/max autocorrelation factors for multivariate spatial statistics. Technical Report 6, Stanford University, 1984.

[Son] www.sony.co.uk.

[TA05] V. Tsagaris and V. Anastassopoulos. Multispectral image fusion for improved RGB representation based on perceptual attributes. *International Journal of Remote Sensing*, 26(15):3241–3254, August 2005.

[UB04a] C. Unsalan and K. L. Boyer. Classifying land development in high resolution panchromatic satellite images using straight line statistics. *IEEE Trans. Geosci. Remote Sensing*, 42(4):907–919, 2004.

[UB04b] C. Unsalan and K. L. Boyer. Classifying land development in high resolution satellite imagery using hybrid structural multispectral features. *IEEE Trans. Geosci. Remote Sensing*, 42(12):2840–2850, 2004.

[Uns06] C. Unsalan. Gradient magnitude based support regions in structural land use classification. *IEEE Geosci. Remote Sensing Letters*, 3:546–550, 2006.

[VGC+93] G. Vane, R. Green, T.G. Chrien, H.T. Enmark, E.G Hansen, and W.M. Porter. The Airborne Visible/Infrared Imaging Spectrometer, (AVIRIS). *Remote Sens. Environ.*, 44(2-3):127–143, May-June 1993.

[vGJMR08] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. LSD: A line segment detector. Technical report, Centre de Mathématiques et de leurs Applications (CMLA), Ecole Normale Superieure de Cachan (ENS-CACHAN), 2008.

[vGJMR10] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. LSD: A fast line segment detector with a false detection control. *IEEE Trans. Pattern Anal. Machine Intell.*, 32(4):722–732, 2010.

[ZW07] P. Zhong and Runsheng Wang. Using combination of statistical models and multilevel structural information for detecting urban areas from a single gray-level image. *IEEE Trans. Geosci. Remote Sensing*, 45(5), 2007.

Appendix I

Annexe 1

AVIRIS hyperspectral image of Moffett Field, California, CA 94035, acquired on August 20, 1992, at the southern end of the San Francisco Bay (Lat1: 37.4708, Lon1: -122, Lat2: 37.4708, Lon2: -122.167) (Ref.: http://aviris.jpl.nasa.gov) [Nas].

Figure I.1: MoffetField2 Hyperspectral Image

AVIRIS hyperspectral image acquired on August 20, 1992 by a NASA ER-2 plane at an altitude of 20000 meters (65000 feet) over Moffett Field, California, CA 94035, (Lat1: 37.45, Lon1: -122, Lat2: 37.45, Lon2: -122.167) at the southern end of the San Francisco Bay. It can be seen on the top of the image is the Moffett Field airport. Of particular interest is the small region of high response in the upper left corner of the larger side. This response is in the red part of the visible spectrum (about 700 nanometers), and is due to the presence of 1-centimeter-long (half-inch) red brine shrimp in the evaporation pond (Ref.: http://aviris.jpl.nasa.gov) [Nas].

Figure I.2: MoffetField3 Hyperspectral Image

Airborne Visible Infrared Imaging Spectrometer (AVIRIS) image of Jasper Ridge Biological Preserve (JRBP) (Lat1:37.3883, Lon1: -122.17, Lat2: 37.4508, Lon2: -122.243) acquired on June 1992 (Ref.: http://aviris.jpl.nasa.gov) [Nas].

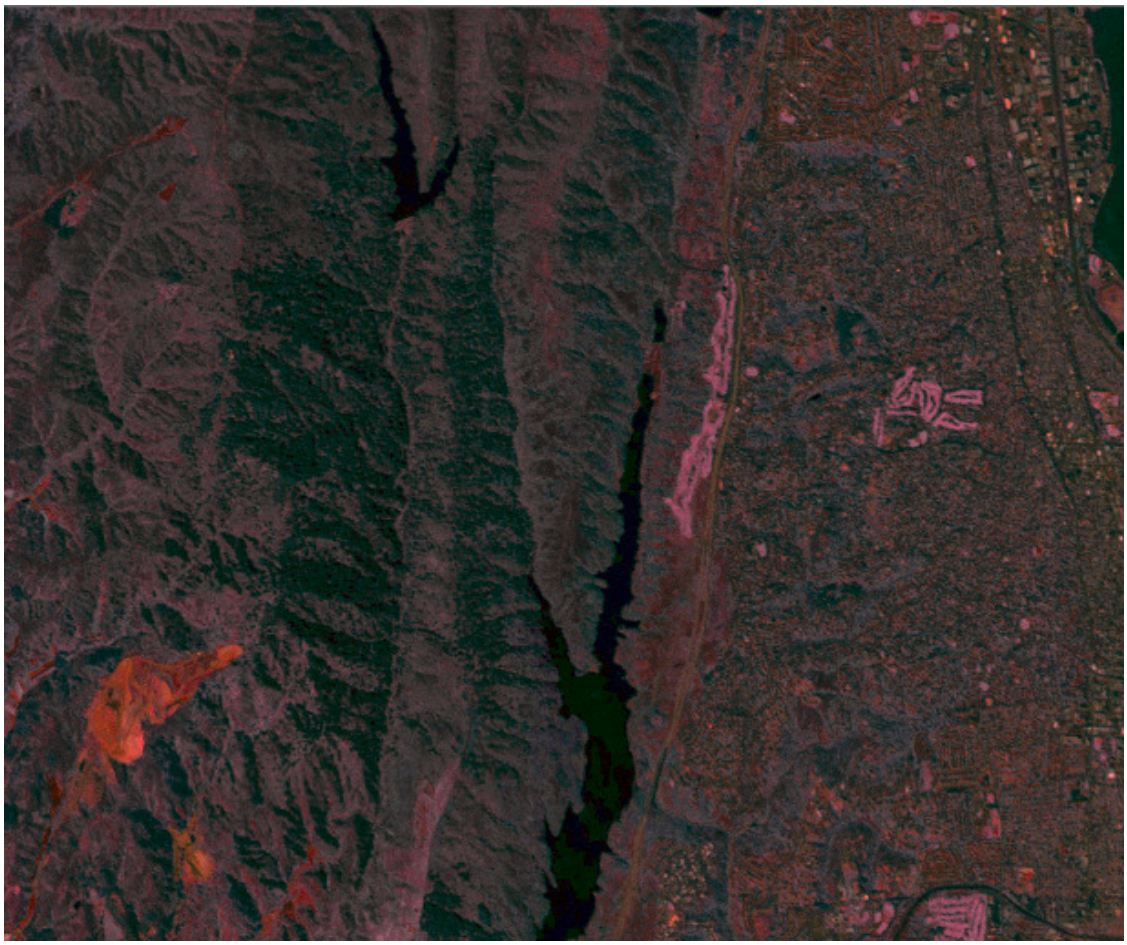


Figure I.3: JasperRidge1 Hyperspectral Image

Airborne Visible Infrared Imaging Spectrometer (AVIRIS) image of Jasper Ridge Biological Preserve (JRBP) (Lat1: 37.375, Lon1: −122.188, Lat2: 37.4375, Lon2: −122.261) acquired on June 1992 (Ref.: http://aviris.jpl.nasa.gov) [Nas].

Figure I.4: JasperRidge2 Hyperspectral Image